Symbolic approaches for explainable artificial intelligence

Matthieu Bellucci

Normandie Université

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN NORMANDIE

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité **INFORMATIQUE**

Préparée au sein de l'**INSA Rouen Normandie**

Approches symboliques pour une intelligence artificielle explicable

Présentée et soutenue par

MATTHIEU BELLUCCI

Thèse soutenue le 01/12/2023

devant le jury composé de :

M. CHRISTOPHE CRUZ	PROFESSEUR DES UNIVERSITÉS - Université de Bourgogne, Dijon	Rapporteur
M. PIERRE MARQUIS	PROFESSEUR DES UNIVERSITÉS - Université d'Artois, Lens	Rapporteur
M. NICOLAS DELESTRE	MAÎTRE DE CONFÉRENCES - INSA de Rouen Normandie	Membre Co-encadrant
MME CLAUDIA FRYDMAN	PROFESSEUR DES UNIVERSITÉS - Université d'Aix-Marseille	Membre
M. NICOLAS MALANDAIN	MAÎTRE DE CONFÉRENCES - INSA de Rouen Normandie	Membre Co-encadrant
MME MARIE-HELENE ABEL	PROFESSEUR DES UNIVERSITÉS - Université de Technologie de Compiègne	Président du jury
MME CECILIA ZANNI-MERK	PROFESSEUR DES UNIVERSITÉS - INSA de Rouen Normandie	Directeur de thèse

Thèse dirigée par **CECILIA ZANNI-MERK** (LABORATOIRE D'INFORMATIQUE DE TRAITEMENT DE L'INFORMATION ET DES SYSTEMES)

MI/IΣ

litis

Abstract

Artificial Intelligence (AI) algorithms are increasingly present in our life and in many industry applications. This widespread adoption is due to the significant progress in performance achieved by machine learning models. These algorithms are being applied to assist decision-makers in sensitive domains such as healthcare, justice or banking. However, the impressive performance of the latest machine learning models come at the expense of understanding their functioning. These models are commonly called "black-boxes" as it is impossible to comprehend their decision process and the influence of each feature on the output. This opacity can hide biased or discriminatory decisions made by these models; decisions that have the potential to impact someone's life. As a result, scientific communities along with legislators, governments and tech companies have started investigating the design of an explainable AI (XAI).

A promising direction for the development of a powerful and explainable AI is the combination of symbolic AI approaches with machine learning models, resulting in neurosymbolic methods. While machine learning models are preferred for their performance, symbolic AI methods are known to be explainable as they exploit human knowledge and logic to make a decision. Furthermore, Semantic Web technologies and especially ontologies have been identified as ideal candidates for the design of explainable AI algorithms. Yet, the current work on neurosymbolic techniques is focused on improving the performance rather than designing explainable AI systems despite the potential of these new architectures to provide explanations. Consequently, this thesis is dedicated to exploring applications of symbolic AI methods to create an explainable AI system.

Beforehand, a study of the terminology of XAI is conducted, as important notions are not clearly defined in the literature. Particularly, the characteristics of explainability and the definition of an explanation are explored in order to ground our contributions. A terminology is introduced that identifies and defines recurring terms of XAI that are ambiguous in the literature. The main finding is that an explanation is an interactive process that determines a set of causes that led to the outcome of an AI system.

Then, we introduce an ontology-based image classifier (OBIC) capable of detecting errors in its predictions. This system exploits an ontology that describes the domain of application to train machine learning models capable of detecting the class of the image and a set of properties defined in the ontology e.g. the texture or color of an object. An inconsistency in the predictions is detected by the ontology and signifies that there was an error in the classification thus helping a user to decide whether to trust the final prediction. The error detection participates in creating a safe AI that users can trust which is the main goal of the XAI field. Moreover, the predictions from the machine learning models are grounded in domain knowledge which facilitates the comprehension and explanation of the prediction. Explanations are an interactive process, hence we built an explanation interface that extracts useful information from this system and formulates adequate explanations.

In order to explain the error detection step from the explainable intelligent system, we need to design an explanation method for ontologies adapted for most users i.e. laypersons, domain experts an AI experts. Among the most popular explanation techniques, counterfactual explanations seem to present many advantages and are being heavily studied to explain machine learning models. We propose a method to generate counterfactual explanations for ontologies that is compatible with most ontologies. It is adapted to explain the functioning of the ontology to laypersons, making this solution ideal to explain the error detection step. It is also suited to assist the ontology designer in the debugging phase by highlighting unexpected inferences caused by design issues.

Finally, we evaluate our contributions on the task of classifying images of musical instruments. An ontology and a dataset are created specifically for this task. The quality and validity of the error detection phase are tested and analyzed. A small scale user study is conducted with domain experts to evaluate the relevance and quality of counterfactual explanations generated by CEO to explain the results of OBIC.

Keywords: *Explainable AI; Explanations; Ontology; Symbolic AI; Semantic Technologies; Machine Learning.*

Résumé

Les algorithmes d'intelligence artificielle (IA) sont de plus en plus présents dans notre vie et dans de nombreuses applications industrielles. Cette adoption généralisée est due aux progrès considérables réalisés par les modèles d'apprentissage automatique en termes de performances. Ces algorithmes sont appliqués pour aider les décideurs dans des domaines sensibles tels que la santé, la justice ou la banque. Cependant, les performances impressionnantes des derniers modèles d'apprentissage automatique se font au détriment de la compréhension de leur fonctionnement. Ces modèles sont communément appelés "boîtes noires" car il est impossible de comprendre leur processus de décision et l'influence de chaque caractéristique sur le résultat. Cette opacité peut cacher des décisions biaisées ou discriminatoires prises par ces modèles ; des décisions qui peuvent avoir un impact sur la vie d'une personne. C'est pourquoi la communauté scientifiques, les législateurs, les gouvernements et les entreprises technologiques ont commencé à étudier la conception d'une IA explicable (XAI).

Une direction prometteuse pour le développement d'une IA puissante et explicable est la combinaison d'approches symboliques de l'IA avec des modèles d'apprentissage automatique, résultant en des méthodes neurosymboliques. Alors que les modèles d'apprentissage automatique sont préférés pour leurs performances, les méthodes d'IA symbolique sont connues pour être explicables car elles exploitent les connaissances et la logique humaines pour prendre une décision. En outre, les technologies du Web Sémantique et en particulier les ontologies ont été identifiées comme des candidats idéaux pour la conception d'algorithmes d'IA explicables. Pourtant, les travaux actuels sur les techniques neurosymboliques se concentrent sur l'amélioration des performances plutôt que sur la conception de systèmes d'IA explicables, malgré le potentiel de ces nouvelles architectures à fournir des explications. Par conséquent, cette thèse est consacrée à l'exploration des applications des méthodes symboliques d'IA pour créer un système d'IA explicable.

Au préalable, une étude de la terminologie de la XAI est menée, car des notions importantes ne sont pas clairement définies dans la littérature. En particulier, les caractéristiques de l'explicabilité et la définition d'une explication sont explorées afin d'ancrer nos contributions sur ces définitions. Une terminologie est introduite qui identifie et définit les termes récurrents de la XAI qui sont ambiguës dans la littérature. La principale conclusion est qu'une explication est un processus interactif qui détermine un ensemble de causes ayant conduit au résultat d'un système d'IA.

Ensuite, nous présentons un classificateur d'images basé sur une ontologie (OBIC) capable de détecter les erreurs dans ses prédictions. Ce système exploite une ontologie qui décrit le domaine d'application pour former des modèles d'apprentissage automatique capables de détecter la classe de l'image et un ensemble de propriétés définies dans l'ontologie, par exemple la texture ou la couleur d'un objet. Une incohérence dans les prédictions est détectée par l'ontologie et signifie qu'il y a eu une erreur dans la classification, ce qui aide l'utilisateur à décider s'il doit faire confiance à la prédiction finale. La détection des erreurs participe à la création d'une IA sûre à laquelle les utilisateurs peuvent faire confiance, ce qui est l'objectif principal du domaine du XAI. En outre, les prédictions des modèles d'apprentissage automatique sont fondées sur la connaissance du domaine, ce qui facilite la compréhension et l'explication de la prédiction. Les explications sont un processus interactif, c'est pourquoi nous avons construit une interface d'explication qui extrait des informations utiles de ce système et formule des explications adéquates.

Afin d'expliquer l'étape de détection des erreurs du système intelligent explicable, nous devons concevoir une méthode d'explication pour les ontologies adaptée à la plupart des utilisateurs, c'est-à-dire les non-initiés, les experts du domaine et les experts en IA. Parmi les techniques d'explication les plus populaires, les explications contrefactuelles semblent présenter de nombreux avantages et font l'objet d'études approfondies pour expliquer les modèles d'apprentissage automatique. Nous proposons une méthode de génération d'explications contrefactuelles pour les ontologies qui est compatible avec la plupart des ontologies. Elle est adaptée pour expliquer le fonctionnement de l'ontologie aux non-initiés, ce qui rend cette solution idéale pour expliquer l'étape de détection des erreurs. Elle est également adaptée pour aider le concepteur de l'ontologie

dans la phase de débogage en mettant en évidence les déductions inattendues causées par des problèmes de conception.

Enfin, nous évaluons nos contributions sur la tâche de classification d'images d'instruments de musique. Une ontologie et un jeu de données sont créés spécifiquement pour cette tâche. La qualité et la validité de la phase de détection des erreurs sont testées et analysées. Une étude utilisateur à petite échelle est menée avec des experts du domaine pour évaluer la pertinence et la qualité des explications contrefactuelles générées par CEO pour expliquer les résultats d'OBIC.

Mots-clés: *IA Explicable; Explications; Ontologie; IA Symbolique; Technologies Sémantiques; Apprentissage Automatique.*

Remerciements

Dans cette section, je souhaite remercier toutes les personnes qui m'ont aidé de près ou de loin à mener à bien cette thèse.

Tout d'abord, je remercie les membres du jury pour le temps qu'ils ont consacré à mes travaux et leurs retours constructifs et bienveillants.

Je tiens à remercier chaleureusement mon équipe d'encadrement de thèse, Nicolas M., Nicolas D. et Cecilia. Je vous remercie pour votre bienveillance, votre patience et votre gentillesse qui m'ont permis de mener à bien cette thèse dans d'excellentes conditions. Je me souviendrais de nos réunions avec les débats animés des deux Nico, déclenchés par mes explications approximatives à propos de mes dernières avancées, avant que Cecilia ne tranche le débat. Je vous remercie infiniment pour tout vos retours, critiques et conseils, qui m'ont permis de fournir le meilleur de moi-même. Vous m'avez poussé à être plus rigoureux dans mes travaux tout en me laissant une grande autonomie et liberté. Je vous suis reconnaissant pour votre implication dans les phases de relecture, avec des retours très constructifs et détaillés, malgré l'aversion pour les longs documents rédigés dans la langue de Shakespeare. Cecilia, je te remercie pour ton dynamisme, ta motivation sans faille et ton optimisme. Tu m'as permis d'y voir plus clair lorsque j'étais perdu et tu as su tirer le meilleur de moi-même (même s'il a fallu quelques fois me mettre la pression en endossant ta redoutable casquette de directrice de thèse). Tu m'as aussi permis de beaucoup voyager, ce qui n'était pas gagné avec le coronavirus (ce coquin de virus, pour paraphraser Renaud). Nicolas M., je te remercie d'avoir toujours su prendre du recul sur mes travaux et donner des retours plus que constructifs, toujours avec le sourire et humour, qui ont rendu les réunions beaucoup plus agréables. Nicolas D., je te remercie également pour ta pédagogie hors norme, ta curiosité et ton enthousiasme constant. Tu as toujours cherché à aller plus loin et titiller ma curiosité en me donnant des pistes ou en me parlant des dernières choses (ou plutôt vidéos) que tu as vues.

Ensuite, je voudrais remercier tous les collègues du LITIS et de l'INSA. Grâce à vous, j'ai passé 3 années dans un cadre accueillant et chaleureux. Je tiens à remercier tout particulièrement Sandra et Brigitte pour leur aide et leur amabilité. C'était à chaque fois un plaisir de vous voir, discuter ou organiser des évènements avec vous. Je remercie tous les collègues/amis doctorants du couloir du 1er étage de Boulingrin et du LITIS en général, je préfère ne pas donner tous les noms de peur d'en oublier. Nous avons passé de supers moments ensemble, à se crier dessus pendant (et après, n'est-ce pas Imane) les parties intenses de LaserGame. Je souhaite remercier Matthias avec qui j'ai partagé un bureau et le rôle de représentant des doctorants. Je crois que l'on a fait une bonne équipe pour redonner un peu de vie dans le labo dans la période post-Covid. Notre organisation était parfaitement équilibrée, tu rédigeais les mails et je les vérifiais, une parfaite répartition des tâches ! Je remercie également Mathieu, membre officiel des meubles du LITIS, pilier des repas à la cafet', qui a d'abord été mon professeur avant de devenir mon collègue et ami. Tu as toujours été de bons conseils, toujours là pour nous aider et nous guider dans les différentes étapes du doctorat.

Je remercie plus généralement le personnel de l'INSA Rouen qui a toujours été amical et humain (comme le veut le slogan). J'y ai passé 8 ans de ma vie et grâce à eux, mon cursus universitaire s'est très bien déroulé, dans d'excellentes conditions.

Nous sortons maintenant du cadre universitaire pour remercier toutes les personnes qui m'ont accompagnées et soutenues pendant mes années de thèse. Je remercie tous mes amis musiciens de l'AMIR, de la SME, d'Opus76, du conservatoire de Rouen et de l'OVVA, avec qui j'ai eu le privilège de faire de la musique pendant ces trois années. J'ai adoré partager autant de moments avec vous, que ce soit en répétition, en concert ou en soirée. Je fais une dédicace toute particulière au Pink Fluffy avec qui j'ai passé mes meilleurs moments musicaux et également à l'Orient Express, meilleur kebab de Rouen (et mon voisin au passage). Dédicace aussi à mes collègues coordinateurs SME et surtout Souha (prononcer Souya) pour toutes les soirées et instants musicaux que l'on a partagés.

Enfin, je remercie particulièrement mes amis et ma famille, qui m'ont encouragés, motivés

et supportés (dans tous les sens du terme). Je vais certainement oublier certaines personnes et je m'en excuse par avance. Merci Alexandre "Zarackai", Frédéric "Monaciello" et Adrien "Rose Strauss" pour nos sessions de jeux et nos visionnages de YTP quotidiens, toujours dans le fun et la bonne humeur ! Merci M. Gouteux et Marine pour avoir toujours été là quand ça n'allait pas, votre hospitalité et votre gentillesse mais aussi nos conversations sur la bourse et l'économie mondiale. Merci messieurs Dassou et Clavel, c'est toujours un plaisir de vous voir et passer du temps avec vous à se remémorer le bon vieux temps, malgré la distance et l'ermitage. Merci Gaétan pour tous ces moments musicaux, les discussions lunaires, les fous rires en répétition et tes collisions avec les sangliers. Merci Manon, pour ta gentillesse et ton dynamisme, ton humour que j'affectionne tant, et ta capacité à monter/démonter avec ma soeur lors de mes nombreux déménagements (je parle de meubles). Merci Robin, pour nos 400 coups, nos délires, ta créativité débordante et pour m'avoir suivi dans tous les projets douteux qui m'ont permis de m'épanouir en dehors de la thèse (je sais que tu es déçu de l'absence de photo de Dominique dans ce manuscrit). Et enfin, merci à Maman, Papa et Marion pour votre présence et votre soutien. Vous m'avez toujours accompagné et aidé dans mes projets, j'ai une chance inouïe d'avoir une telle famille, prête à faire 700km pour m'aider à déménager (ai-je mentionné que j'ai beaucoup déménagé durant ces années ?) ou qui m'ont généreusement proposé de relire ce manuscrit de thèse. Je vous suis infiniment reconnaissant pour toutes les opportunités que vous m'avez donné.
Finalement, merci à toi, lecteur de cette thèse, pour l'intérêt que tu portes à mes travaux !

Bravo aux remerciements !

Contents

Contents vii

List of Figures xi

1 Synthèse de la thèse en français 1
- 1.1 Introduction . . . 1
  - 1.1.1 Introduction à l'IA Explicable . . . 3
  - 1.1.2 Motivations et contributions . . . 5
  - 1.1.3 Publications . . . 7
- 1.2 État de l'art . . . 8
  - 1.2.1 Définition d'une explication . . . 8
  - 1.2.2 Évaluation d'une explication . . . 10
  - 1.2.3 Aperçu des méthodes d'explication . . . 10
  - 1.2.4 Conclusion . . . 13
- 1.3 Contributions . . . 14
  - 1.3.1 Terminologie du XAI . . . 14
  - 1.3.2 Un système intelligent explicable fondé sur une ontologie pour classer les images . . . 15
  - 1.3.3 Explications contrefactuelles pour les ontologies . . . 16
- 1.4 Conclusions et travaux futurs . . . 18
  - 1.4.1 Conclusion . . . 18
  - 1.4.2 Travaux futurs . . . 19

2 Introduction 23
- 2.1 Context . . . 23
- 2.2 Introduction to Explainable AI . . . 25
- 2.3 Motivations and outline . . . 27
  - 2.3.1 Motivations . . . 27
  - 2.3.2 Contributions and outline . . . 28
- 2.4 Publications . . . 29

3 Background on XAI and Symbolic AI 31
- 3.1 Background on XAI . . . 31
  - 3.1.1 Defining an explanation . . . 32
  - 3.1.2 Designing explainability methods . . . 35
  - 3.1.3 Evaluating XAI . . . 38
  - 3.1.4 Limitations . . . 47
- 3.2 Background on ontologies . . . 49
  - 3.2.1 Ontologies and their applications . . . 50
  - 3.2.2 Description logics . . . 52
  - 3.2.3 The Web Ontology Language (OWL) . . . 53
  - 3.2.4 Discussion . . . 56

4 The XAI terminology **57**
4.1 Literature review . . . 58
4.1.1 Interpretability and transparency . . . 58
4.1.2 Explainability and explanations . . . 61
4.1.3 Responsible AI terms . . . 62
4.2 A terminology for a contextualized XAI . . . 64
4.2.1 System terminology . . . 65
4.2.2 Explanation terminology . . . 70
4.2.3 Formalization of XAI explanations . . . 72
4.3 Conclusion . . . 73

5 An ontology-based explainable intelligent system to classify images **75**
5.1 Literature review . . . 76
5.1.1 Error detection . . . 77
5.1.2 Explainable image classifiers . . . 78
5.1.3 Explainable neurosymbolic methods . . . 82
5.1.4 Explanation interfaces . . . 83
5.1.5 Discussion . . . 84
5.2 OBIC: explainable ontology-based image classifier . . . 85
5.2.1 Ontology requirements . . . 87
5.2.2 Training phase . . . 87
5.2.3 Inference phase . . . 90
5.3 Explanations with OBIC . . . 92
5.3.1 Extraction of the explanations . . . 92
5.3.2 Design of the explanation interface . . . 94
5.4 Conclusion . . . 96

6 Counterfactual explanations for ontologies **99**
6.1 Literature review . . . 99
6.1.1 Counterfactual explanations . . . 100
6.1.2 Explaining ontologies . . . 106
6.1.3 Similarity metrics for individuals . . . 106
6.2 Counterfactual explanations for ontologies . . . 109
6.3 The CEO method . . . 114
6.3.1 Exploring the counterfactuals space . . . 116
6.3.2 Computing metrics . . . 121
6.4 Validation . . . 122
6.4.1 Test cases and results . . . 123
6.4.2 Analysis . . . 124
6.5 Conclusion . . . 126

7 Experiments **127**
7.1 Evaluation task . . . 127
7.1.1 The musical instruments ontology . . . 128
7.1.2 The musical instruments dataset . . . 129
7.2 Experiments on OBIC . . . 130
7.2.1 Methodology . . . 130
7.2.2 OBIC implementation . . . 131
7.2.3 Results . . . 132
7.2.4 Analysis . . . 135
7.3 Experiments on the counterfactual explanations . . . 138
7.3.1 Methodology . . . 139
7.3.2 Results . . . 140

7.3.3 Analysis 142
7.4 Conclusion 144

8 Conclusion and future work **147**

A CEO: Size of the search space **151**

B Definition of the musical instruments ontology **153**

List of Figures

1.1 Un système d'IA explicable tel que décrit par la DARPA [15] 7
1.2 Diagramme d'une explication 9
1.3 Exemple d'explication de classification d'image par importance des variables, par plusieurs méthodes populaires [52] 12
1.4 Diagramme du fonctionnement d'OBIC 15

3.1 Diagram of an explanation 33
3.2 A comparison of two taxonomies of XAI methods 36
3.3 Example of a semantic network that describes a truck [114]. 49
3.4 The three levels of ontologies [122] 51
3.5 The Semantic Web Stack [126] 51

4.1 Evolution of the number of total publications whose title, abstract and/or keywords contained the terms in the legend [28]. 59
4.2 The ontology design pattern to define explanations [65]. 72
4.3 ODP to define explanation for explanations in XAI 73

5.1 An explainable AI system as described by DARPA [15] 76
5.2 Saliency maps for some common methods compared to an edge detector [52]. . . . 79
5.3 The explanation output from the method of Pintelas et al. [55]. 81
5.4 Diagram of the functioning of OBIC 86
5.5 Design pattern from [179] used by OBIC 86
5.6 Output of the `:hasMechanism` classifier. 88
5.7 Illustration of the labeling and training process. 90
5.8 Two examples of the explanation interface 95

6.1 An IKG representing the customer *Alice*, classified as `Denied` 111
6.2 Graph representation of the search space Ω. 115
6.3 Process of exploring the search space 117
6.4 Process of generating ancestors for one individual. 118
6.5 Process of generating descendants for one individual. 120
6.6 Bar plot of the detailed execution time for each example. 125

7.1 Classes and object properties hierarchies of the musical instruments ontology 128
7.2 Class distribution in the musical instruments dataset 130
7.3 Illustration of the thresholds in OBIC 131
7.4 Average number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) for each threshold 133
7.5 Average precision, recall, F1-score and FPR for each threshold 133
7.6 Evolution of classification scores when removing `:hasMechanism` classifier. 134
7.7 FPR-Precision curve with and without `:hasMechanism` classifier 135
7.8 Description and the first four explanations of the first case of the survey 140
7.9 Input image of the first case of the survey. 140

8.1 Diagram of a framework that generalizes OBIC to any type of task and data 150

Chapter 1

Synthèse de la thèse en français

Le sujet de cette thèse se situe à l'intersection de différents domaines, mais se concentre principalement sur l'IA eXplicable (XAI). Nous présentons le contexte de cette thèse, le domaine du XAI et les motivations de cette thèse dans la Section 1.1. Ensuite, nous proposons une synthèse de l'état de l'art du XAI dans la Section 1.2. Puis, nous décrivons les contributions de cette thèse dans la Section 1.3. Enfin, nous concluons cette synthèse et discutons des travaux futurs dans la Section 1.4.

1.1 Introduction

L'intelligence Artificielle (IA) moderne est apparue à la fin des années 1940 grâce aux progrès de l'informatique et de la logique formelle. Au cours de son histoire, le domaine de l'IA a connu trois "étés" et deux "hivers" qui désignent les périodes où le financement de la recherche en IA était soit abondant, soit limité. Nous nous trouvons actuellement dans le troisième été de l'IA, qui est dominé par l'apprentissage automatique et plus particulièrement le Deep Learning [1]. En effet, de la vision par ordinateur au traitement du langage naturel, il n'y a pas un domaine qui n'a pas été affecté par les progrès dans les réseaux de neurones. Cela a conduit à une forte adoption par les entreprises dans de multiples domaines tels que la santé, la justice, l'industrie automobile ou même l'art [2]. Les réseaux de neurones artificiels sont des modèles flexibles qui peuvent approximer des fonctions mathématiques, sous condition d'avoir les paramètres appropriés. Ces paramètres sont généralement trouvés à l'aide de l'algorithme de rétropropagation qui utilise la descente de gradient pour les ajuster après chaque observation d'un nouveau point de données. Toutefois, cette méthode est coûteuse en calcul et nécessite une grande quantité de données; notamment pour les réseaux de neurones récents, tels que GPT-3, qui comptent des centaines de milliards de paramètres [3]. Le potentiel des réseaux de neurones a été débloqué par la récente augmentation de la puissance de calcul et la disponibilité de grandes quantités de données, ce qui a conduit à ce troisième été de l'IA qui a commencé vers 2012 [1]. Cependant, les réseaux de neurones ont plusieurs inconvénients qui rendent leur application indésirable dans certains domaines. L'une de ces limitations est l'incapacité d'expliquer la décision prise par ce type d'algorithmes. Par conséquent, leur application est mal perçue dans les domaines sensibles qui ont un impact direct sur des vies humaines. Ce problème d'explicabilité a également été rencontré lors du dernier été de l'IA, qui a vu l'essor des systèmes experts. Néanmoins, les systèmes experts sont des algorithmes d'IA symbolique qui ont l'avantage d'utiliser les connaissances humaines et le raisonnement déductif pour leur processus de décision. Ils sont donc plus facilement explicables que les réseaux de neurones. Cette propriété des systèmes experts a incité les chercheurs à explorer leur combinaison avec les réseaux de neurones pour résoudre le problème d'explicabilité de ces derniers.

Les systèmes experts étaient aussi populaires dans les années 1970 que les réseaux de neurones le sont aujourd'hui. Ils sont issus des progrès de l'IA symbolique, une classe d'algorithmes d'IA qui manipulent des symboles qui peuvent être compréhensibles par les humains. Comme les réseaux de neurones, les systèmes experts ont rapidement été utilisés dans de nombreux do-

maines pour une variété de tâches. On estime que plus des deux tiers des entreprises du classement Fortune 1000 appliquaient des systèmes experts dans leurs activités quotidiennes dans les années 1980 [4]. Les systèmes experts, et plus généralement les systèmes fondés sur la connaissance, appliquent une logique formelle aux connaissances humaines pour prendre une décision ou faire une prédiction. Deux composantes principales interagissent pour résoudre une tâche: une base de connaissances et un moteur d'inférence [5]. La base de connaissances contient les connaissances spécifiques du domaine nécessaires à la résolution des problèmes. Elle utilise des formes de représentation de la connaissance telles que des règles ou des réseaux sémantiques. Ces connaissances sont acquises avec l'aide d'experts du domaine qui collaborent avec un ingénieur pour encoder leurs connaissances et leur expérience. Ensuite, le moteur d'inférence est capable de raisonner et d'interpréter les règles et les connaissances contenues dans la base de connaissances. Sa tâche consiste à trouver des chemins logiques dans la forêt de règles pour arriver à une conclusion. L'intérêt soudain pour les systèmes experts est dû à leur capacité à reproduire automatiquement le processus de décision d'un expert humain. Ils ont permis aux entreprises d'économiser du temps et de l'argent sur des tâches répétitives mais très spécifiques. Le premier exemple d'adoption réussie d'un système expert dans l'industrie est celui de XCON [6] qui a considérablement réduit le processus de personnalisation d'un ordinateur de 90 jours à 90 minutes [1].

Le dénominateur commun entre l'apprentissage automatique et l'IA symbolique est leur objectif de reproduire le raisonnement humain. En effet, les systèmes experts et les réseaux de neurones imitent un sous-ensemble de l'intelligence humaine : la capacité de raisonner et de déduire des faits pour les systèmes experts et la capacité d'apprendre et d'induire de nouvelles connaissances pour les réseaux de neurones. Toutefois, comme nous l'avons déjà mentionné, un comportement humain clé est la capacité d'expliquer, en particulier pour les décisions importantes Selon plusieurs chercheurs qui ont travaillé sur les systèmes experts, un système d'IA doit être capable d'expliquer son processus de décision pour assurer l'acceptation de l'utilisateur [7]. Grâce à la nature symbolique des systèmes experts, les explications sont faciles à générer et consistent généralement à retracer les étapes logiques qui ont conduit à une décision. À l'inverse, expliquer les réseaux de neurones est une tâche nettement plus difficile car ils n'utilisent pas de symboles compréhensibles par l'homme. De plus, le fonctionnement des réseaux de neurones est beaucoup plus complexe et nécessite des connaissances mathématiques avancées pour comprendre les phases d'apprentissage et d'inférence. Ce manque d'explicabilité peut avoir des conséquences désastreuses si l'on se fie aveuglément à leurs décisions. En effet, lorsqu'ils apprennent à partir de données biaisées, les modèles de Deep Learning renforcent ces biais et prennent par conséquent des décisions discriminatoires [8]. Motivé par ces questions ainsi que par l'application de la réglementation RGPD par l'Union européenne qui promeut un "droit à l'explication" [9], le domaine de recherche de l'IA explicable gagne rapidement en popularité et est exploré par les universitaires et les industries.

Outre le problème de l'explicabilité, les réseaux de neurones présentent d'autres défauts. À savoir un manque de robustesse et la nécessité de disposer de quantités massives de données et de puissance de calcul [10]. En leur temps, les systèmes experts ont également souffert de défauts intrinsèques tels que leur incapacité à gérer le raisonnement avec incertitude et la difficulté d'acquérir des connaissances expertes suffisantes [1]. Ces problèmes ont conduit à la disparition des systèmes experts et au début d'un hiver de l'IA. Les réseaux de neurones risquent de connaître le même sort si aucune solution n'est trouvée. Heureusement, des chercheurs ont récemment plaidé en faveur de l'hybridation de l'IA symbolique et des réseaux de neurones afin de surmonter les défauts des deux approches. Cette idée est également motivée par le fait que l'intelligence humaine utilise à la fois la déduction et l'induction. Afin de la reproduire, il semble pertinent que les systèmes d'IA soient également capables de faire les deux. Le domaine de recherche qui cherche à créer des méthodes d'IA hybrides combinant les réseaux neuronaux et l'IA symbolique est appelé IA neurosymbolique.

Dans l'histoire de l'IA, les hivers ont été causés par l'incapacité à remédier aux limites des

méthodes populaires. La communauté de l'IA craint un troisième hiver de l'IA qui serait provoqué par l'absence de solutions aux problèmes susmentionnés du Deep Learning. Les domaines de l'IA explicable et de l'IA neurosymbolique sont de nouveaux domaines qui cherchent à surmonter ces limitations et éviter le sort des deux derniers étés de l'IA. La jeunesse relative de ces domaines signifie que de nombreuses directions de recherche doivent encore être explorées. Par conséquent, ces domaines ne sont pas normalisés, en particulier dans le domaine du XAI où les notions fondamentales ne sont pas clairement définies de manière consensuelle. De plus, le domaine du XAI est confronté à des difficultés qui dépassent le cadre de l'informatique. En particulier, la définition et l'évaluation d'une explication est un problème multidisciplinaire, incluant les sciences sociales, la psychologie ou même la philosophie. Malgré ces difficultés, la recherche dans le XAI se développe rapidement. Un grand nombre de méthodes ont déjà été créées, mais il n'existe toujours pas de méthode normalisée pour les évaluer et les comparer. Enfin, le XAI et l'IA neurosymbolique semblent suivre des voies distinctes avec des objectifs différents. L'explicabilité est généralement mentionnée comme une application possible des méthodes d'IA neurosymbolique, mais elle est rarement explorée et évaluée. Comme le suggère le titre de cette thèse, elle vise à explorer l'utilisation de méthodes symboliques et neurosymboliques pour répondre aux défis actuels du domaine du XAI.

1.1.1 Introduction à l'IA Explicable

L'IA explicable est un domaine qui a été créé en réponse au besoin général d'explicabilité dans l'IA ainsi qu'à l'opacité manifeste des modèles actuels d'apprentissage automatique. Fournir des explications satisfaisantes pour chaque utilisateur est un défi car chaque utilisateur a des attentes, des croyances, des connaissances et des besoins différents. La conception d'une explication ne dépend pas seulement de la personne à qui l'on explique, mais aussi des objectifs de l'explicateur et du contexte général [11]. Par exemple, un médecin n'explique pas un diagnostic de la même manière, qu'il s'adresse à un patient, à un étudiant ou à un collègue. Ils cherchent à gagner la confiance et l'acceptation du patient, à transmettre des connaissances à l'étudiant et à justifier le diagnostic au collègue. Chaque explication sera donc construite différemment pour atteindre chaque objectif. Puisque le XAI produit des méthodes pour générer des explications, il est crucial de comprendre les différents objectifs que le XAI cherche à atteindre pour mieux comprendre le paysage du XAI et ses enjeux actuels.

Deux objectifs principaux d'une explication ont été identifiés par les chercheurs: permettre à l'utilisateur de comprendre un système et d'établir une relation de confiance avec lui [12, 13, 14]. Le programme de recherche de la DARPA sur le XAI [15] est souvent considéré comme le point de départ du XAI moderne et utilisé comme référence. L'objectif de ce programme est de créer une suite de techniques d'apprentissage automatique pour produire des modèles explicables qui, combinés avec des méthodes d'explication, permettent aux utilisateurs de comprendre, faire confiance de manière appropriée et de gérer efficacement la nouvelle génération de systèmes d'IA. [16]. La confiance est donc considérée comme un objectif fondamental du XAI. De plus, les récents développements en IA ont soulevé de nouveaux problèmes qui peuvent être résolus par le XAI. Très récemment, de nouveaux agents conversationnels ont été créés et mis à la disposition du public. En particulier, Meta (l'entreprise qui possède des réseaux sociaux tels que Facebook ou Instagram) a lancé Galactica, un modèle de langage entraîné sur des articles scientifiques, capable de stocker, de combiner et de raisonner sur des connaissances scientifiques [17]. Bien qu'il ait montré des résultats prometteurs sur les tests de performances classiques, les utilisateurs ont constaté que Galactica ne générait que de fausses informations, de manière confiante [18]. Le public n'a pas tardé à dénoncer cette IA comme étant contraire à l'éthique, voire dangereuse. Au cours de la même période, un modèle similaire a été mis à disposition: ChatGPT. Contrairement à Galactica, ChatGPT a connu un succès et une popularité incroyables, atteignant des millions d'utilisateurs quotidiens en quelques jours [19]. Les bonnes performances de ce modèle ont conduit à une utilisation discutable de celui-ci, par exemple plusieurs articles de recherche ont été publiés avec ChatGPT comme co-auteur [20]. En conséquence, plusieurs questions éthiques concernant l'IA et

son utilisation ont été soulevées et des mesures ont été prises pour éviter les utilisations abusives. Les créateurs de ChatGPT avaient déjà mis en place un système de sécurité pour prévenir ces problèmes, qui détecte et filtre le contenu indésirable [21]. Malgré ces efforts, les utilisateurs ont réussi à trouver des moyens de contourner ce système de sécurité pour générer des contenus nuisibles, ce qui a mis en évidence le manque de robustesse des modèles d'apprentissage profond. Enfin, outre les questions éthiques soulevées par l'utilisation de l'IA, ChatGPT a également soulevé des questions sur le développement de l'IA. En effet, le magazine d'information TIME a révélé qu'afin d'entraîner le système de sécurité, des travailleurs ont été embauchés pour annoter manuellement des données indésirables, les exposant ainsi à des contenus particulièrement violents [22]. Avec ces développements récents, il apparaît que pour que les modèles d'IA soient dignes de confiance, l'explicabilité et l'interprétabilité ne suffisent pas. D'autres exigences doivent être satisfaites pour garantir que les systèmes d'IA sont éthiques et responsables, tant dans leur développement que dans leur utilisation.

Conscientes de ces problèmes croissants et de leurs conséquences néfastes, plusieurs entités allant des entreprises privées aux gouvernements ont proposé des lignes directrices ou des principes pour le développement et l'utilisation d'une IA responsable, éthique ou digne de confiance [23, 24, 25]. Fjeld et al. [26] ont examiné 36 documents proposant des lignes directrices pour la conception d'une IA responsable, provenant de la société civile, de gouvernements, d'organisations intergouvernementales et du secteur privé. Ils ont identifié des tendances et des thèmes qui sont mentionnés dans la majorité des 36 documents. Dernièrement, les documents ont convergé vers des principes clés qui constituent le "noyau normatif" de l'IA responsable. Dans une récente revue de l'IA responsable, Mikalef et al. [27] proposent les principes suivants et leurs descriptions respectives:

Equité Les systèmes d'IA doivent permettre l'inclusion et la diversité et ne pas conduire à des résultats discriminatoires.

Transparence Les systèmes d'IA doivent être ouverts et transparents en ce qui concerne les processus et les résultats, et faciliter la traçabilité, l'explicabilité et la communication avec les utilisateurs.

Responsabilité Les systèmes d'IA doivent être développés en tenant compte de la responsabilité et de l'obligation de rendre compte de leurs résultats dans le respect de l'éthique et des principes.

Robustesse et sécurité Les systèmes d'IA doivent être développés selon une approche préventive des risques et de manière à ce qu'ils se comportent comme prévu tout en minimisant les dommages involontaires et inattendus.

Gouvernance des données Les systèmes d'IA doivent garantir qu'une gouvernance adéquate des données couvre la qualité et l'intégrité des données tout au long de leur cycle de vie.

Lois et réglementations Les systèmes d'IA doivent respecter les lois et réglementations qui régissent leur fonctionnement.

Supervision humaine Les systèmes d'IA doivent générer des avantages tangibles pour les personnes et toujours rester sous le contrôle de l'homme.

Bien-être sociétal et environnemental Les systèmes d'IA doivent promouvoir la responsabilité écologique et sociale, la durabilité et ne pas causer de dommages.

Bien que le XAI semble être une solution exclusivement axée sur la transparence, il peut également être appliqué à d'autres principes. Parmi les 32 principes identifiés par Fjeld et al. [26], 28 principes incluent explicitement le XAI comme une composante cruciale selon Arrieta et al. [28]. En effet, les explications peuvent être utilisées pour atteindre différents objectifs. Comprendre quelles caractéristiques ont été utilisées pour faire une prédiction peut indiquer si le système d'IA

est discriminatoire ou biaisé, ce qui est une exigence pour une IA équitable. De même, les explications facilitent l'audit d'un système d'IA et le signalement des effets négatifs, ce qui est nécessaire pour garantir la responsabilité. Ce nouvel idéal d'une IA responsable utilise le XAI comme principal moyen d'atteindre ses objectifs. Par conséquent, les objectifs du XAI ont été élargis et de nouvelles directions de recherche ont été identifiées. Entre-temps, d'autres objectifs pour le XAI ont été discutés par Adadi et Berrada [29], qui se recoupent avec les objectifs de l'IA responsable. À savoir, expliquer pour justifier, expliquer pour contrôler, expliquer pour améliorer et expliquer pour découvrir. Les explications visant à justifier et à contrôler englobent presque entièrement ce qui a été décrit précédemment. En effet, la justification garantit l'équité et l'auditabilité, ce qui permet d'instaurer la confiance, tandis que l'explication pour le contrôle permet de détecter et de prévenir les erreurs et les défaillances du système, garantissant ainsi la robustesse et la sécurité. Les explications visant à améliorer et à découvrir ne contribuent pas directement à l'IA responsable ni même à l'instauration de la confiance, mais elles peuvent conduire à de nouveaux progrès significatifs dans l'IA si elles sont poursuivies. Selon Adadi et Berada [29], un modèle qui peut être expliqué et compris est un modèle qui peut être facilement amélioré. Ils soutiennent que le XAI pourrait être le fondement d'une amélioration continue entre l'homme et la machine. Alors qu'expliquer pour améliorer traite de la manière dont les humains peuvent améliorer les machines, expliquer pour découvrir est l'inverse. Les modèles d'apprentissage automatique apprennent à partir de données ; ils peuvent donc découvrir de nouvelles connaissances, observer de nouvelles corrélations inconnues de l'homme. Par exemple, les systèmes d'IA basés sur l'apprentissage par renforcement excellent désormais dans des jeux comme les échecs ou le jeu de Go. Il serait souhaitable de comprendre les stratégies apprises, afin d'accroître les connaissances humaines. Plus généralement, avec l'application des algorithmes d'apprentissage à différents domaines scientifiques, la découverte des connaissances apprises par ces modèles pourrait conduire à des percées scientifiques.

Selon le programme XAI de la DARPA, la recherche en XAI peut être organisée en trois domaines représentatifs des défis actuels du XAI [30]:

1. Comprendre la psychologie de l'explication en résumant, en étendant et en appliquant les théories psychologiques de l'explication.

2. Le développement de nouvelles méthodes de XAI pour l'apprentissage automatique et les techniques d'explication pour générer des explications efficaces.

3. L'évaluation des nouvelles techniques de XAI dans deux domaines: l'analyse de données et l'autonomie.

Dans cette thèse, nous explorons des solutions à ces défis en utilisant des approches symboliques, en particulier les technologies du Web sémantique qui ont été identifiées comme prometteuses dans la littérature.

1.1.2 Motivations et contributions

Ce manuscrit présente nos contributions qui visent à explorer de nouvelles solutions pour résoudre le problème de l'explicabilité. Dans cette section, nous discutons de nos motivations pour cette thèse puis nous exposons le contenu de nos contributions.

Motivations

Le principal problème qui se pose dans tous les aspects du XAI est l'absence générale de consensus. En effet, comme le montrent les défis énoncés dans la section précédente, la définition et l'évaluation d'une explication font encore l'objet de débats. Par conséquent, la terminologie du XAI est le premier aspect à souffrir de l'absence de consensus. Nous avons observé des termes identiques ayant des définitions différentes et des termes différents ayant des définitions similaires. Par exemple, les termes “explicabilité” et “interprétabilité” sont parfois définis comme

des synonymes et d'autres fois définis différemment, bien qu'ils fassent partie des termes les plus importants du XAI. De plus, les auteurs ne définissent pas systématiquement les termes utilisés dans leurs articles, ce qui accroît la confusion générale dans la terminologie. Ce problème a des conséquences sur la définition, la conception et l'évaluation des explications, tout en rendant le domaine du XAI particulièrement difficile à comprendre pour les novices.

En relation avec le problème de la terminologie, l'identification des critères pertinents pour évaluer les méthodes de XAI et les explications font l'objet de débats. Les universitaires semblent partager le même point de vue sur les critères qui représentent la qualité de l'explication. Cependant, il n'y a pas de consensus sur les noms, les définitions et les formules mathématiques correspondantes. Par ailleurs, les explications sont un processus social qui implique une part de subjectivité dans l'évaluation. Pourtant, l'évaluation des méthodes XAI se limite principalement à des mesures objectives. Plusieurs revues de la littérature sur l'évaluation des méthodes de XAI ont souligné la rareté des évaluations par des sujets humains, principalement en raison de leur coût et de la difficulté à les mettre en place. Les quelques études sur des utilisateurs existantes ont confirmé que la qualité d'une explication dépend de l'utilisateur et du contexte.

Néanmoins, la communauté du XAI s'accorde à diviser les méthodes en deux catégories : *post hoc* et *ante hoc*. Les méthodes *post hoc* sont idéales pour expliquer les algorithmes "boîte noire", tandis que les méthodes *ante hoc* exploitent la nature interprétable de certains algorithmes d'IA pour générer une explication. Plusieurs auteurs plaident en faveur de l'utilisation de méthodes *ante hoc* et donc de modèles interprétables. Ils affirment que les méthodes *post hoc* manquent de robustesse et de fidélité, ce qui pourrait conduire à des résultats contre-productifs. Inversement, les modèles interprétables "traditionnels" (par exemple, les modèles linéaires, les arbres de décision ou les règles) sont généralement moins performants que leurs équivalents opaques, en particulier lorsqu'ils traitent des données de grande dimension. Même dans les cas où les modèles interprétables atteignent les performances des modèles opaques, la haute dimensionnalité entraîne une augmentation de la complexité et donc une diminution de l'interprétabilité du modèle. Les approches neurosymboliques sont une réponse à ce problème car elles combinent l'interprétabilité des approches symboliques avec les performances de pointe des modèles d'apprentissage automatique. Parallèlement, des modèles auto-explicables sont en cours de développement avec le même objectif de fournir des performances et une interprétabilité élevées. De tels modèles interprétables nécessitent toujours la génération d'une explication adaptée à l'utilisateur. À notre connaissance, il n'existe pas de système d'IA explicable capable de prédire et de générer des explications adaptées à l'utilisateur. Une architecture générique d'un système intelligent explicable (XIS) a été proposée par la DARPA pour orienter la recherche en XAI [15] et est représentée dans la Figure 1.1. Comme les modèles standards d'apprentissage automatique, il nécessite des données et un processus d'apprentissage pour entraîner le modèle. À la différence des modèles standars, le modèle issu de ce processus permet la génération d'explications à propos de ses décisions et de son fonctionnement global. Nous avons vu qu'une explication est un processus interactif, une interface d'explication est donc ajouté au système d'IA pour répondre aux questions des utilisateurs. Les modèles auto-explicables et neurosymboliques sont les plus proches de cette architecture, mais ils n'incluent pas l'une des parties les plus importantes, à savoir le système qui génère et présente les explications à l'utilisateur.

Ces observations motivent la conception d'un XIS complet qui reproduit la globalité de l'architecture proposée par la DARPA. Cette conception doit suivre un ensemble de bonnes pratiques afin d'éviter les pièges détectés dans la littérature, à savoir la terminologie ambiguë et le manque d'évaluation adéquate. De plus, nous observons que l'objectif d'expliquer pour contrôler ou améliorer est mal représenté dans le paysage actuel du XAI. Par conséquent, le XIS devrait être capable de confirmer ou infirmer ses prédictions et d'expliquer pourquoi. Nous émettons l'hypothèse qu'une approche neurosymbolique pourrait être utilisée à cette fin. En particulier, nous avons l'intention d'explorer la combinaison de modèles d'apprentissage automatique avec les technologies du Web Sémantique (par exemple, les ontologies ou les graphes de connaissances), car ces dernières ont été identifiées comme des candidats idéaux pour résoudre les

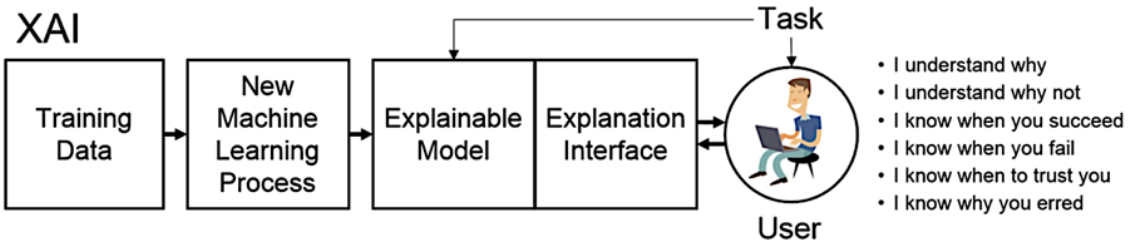


Figure 1.1: Un système d'IA explicable tel que décrit par la DARPA [15]

défis actuels du XAI [31]. Il a été mentionné que les modèles interprétables devraient être associés à des méthodes d'explication qui tirent parti de leur nature interprétable. C'est pourquoi une méthode d'explication spécifique au modèle neurosymbolique que nous proposons sera également développée dans cette thèse.

Contributions

Cette thèse présente trois contributions principales.

1. Une terminologie du XAI, conçue pour éliminer toute ambiguïté dans les définitions tout en restant compatible avec la majorité de la littérature. Cette terminologie contient la définition et la composition d'une explication ainsi que la définition de termes récurrents dans la littérature sur le XAI qui qualifient les systèmes d'IA.

2. La conception d'un XIS basé sur l'architecture de la DARPA [15]. Ce XIS est décomposé en deux parties: un nouveau modèle neurosymbolique pour la classification et une interface d'explication qui tire parti de ce modèle pour générer des explications adéquates. Le modèle neurosymbolique utilise une ontologie pour créer et entraîner des modèles d'apprentissage automatique et vérifier la cohérence des prédictions faites par ces modèles. Il est donc capable d'avertir l'utilisateur lorsqu'une prédiction n'est pas cohérente avec les connaissances expert et d'en expliquer les raisons. Une interface d'explication est ensuite développée pour présenter les résultats du XIS à l'utilisateur.

3. Une méthode pour générer des explications contrefactuelles pour les ontologies. Ce mode d'explication est particulièrement adapté pour expliquer la prédiction et la détection de la cohérence du XIS susmentionné. Il est conçu pour être applicable à la plupart des ontologies existantes en tant qu'outil de débogage. Cette méthode s'inspire de méthodes existantes pour générer de telles explications pour l'apprentissage automatique. Elle utilise les mêmes principes, mais plusieurs problèmes spécifiques aux ontologies se posent et sont résolus dans cette contribution.

Enfin, le XIS ainsi que la méthode d'explication contrefactuelles pour les ontologies sont évalués dans un même cadre expérimental. Une première partie de cette évaluation mesure la capacité du XIS à détecter une erreur faite par les modèles d'apprentissage automatique. La seconde partie est une étude utilisateur qui cherche à déterminer la qualité et la pertinence des explications contrefactuelles qui cherchent à expliquer les prédictions du XIS.

1.1.3 Publications

Les travaux suivants ont été publiés dans des conférences et des revues évaluées par des pairs au cours de cette thèse :

- **Bellucci M.**, Delestre N., Malandain N. and Zanni-Merk C., (2021), *"Towards a terminology for a fully contextualized XAI"*. In Proceedings of the 25th International Conference

on Knowledge-Based and Intelligent Information and Engineering Systems, KES 2021, 8-10 September 2021, Szczecin, Poland. DOI: 10.1016/j.procs.2021.08.025

- **Bellucci M.**, Delestre N., Malandain N. and Zanni-Merk C., (2022), *"Une terminologie pour une IA explicable contextualisée"*. In Conférence francophone sur l'Extraction et la Gestion des Connaissances (EGC 2022) 2022 as part of EXPLAIN'AI Workshop, 24-28 January 2022, Blois, France.

- **Bellucci M.**, Delestre N., Malandain N. and Zanni-Merk C., (2022), *"Ontologies to build a predictive architecture to classify and explain"*. In the European Semantic Web Conference 2022 (ESWC 2022) as part of the Deep Learning meets Ontologies and Natural Language Processing (DeepOntoNLP) Workshop, May 29 - June 2 2022, Hersonissos, Greece.

- **Bellucci M.**, Delestre N., Malandain N. and Zanni-Merk C., (2022), *"Combining an explainable model based on ontologies with an explanation interface to classify images"*. In Proceedings of the 26th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, KES 2022, 7-9 September 2022, Verona, Italy. DOI: 10.1016/j.procs.2022.09.298

1.2 État de l'art

Dans cette section, nous étudions l'état de l'art du domaine de l'IA explicable. Nous discutons des problématiques de la définition et l'évaluation d'une explication. Nous proposons par la suite un aperçu des méthodes d'explication développées par la communauté du XAI.

1.2.1 Définition d'une explication

Une explication est un ensemble d'informations pertinentes, accompagnées d'un certain type de raisonnement, qui permettra à une personne de comprendre les raisons d'un phénomène. La Figure 1.2 représente les acteurs et composantes d'une explication, inspiré du diagramme de Cabitza et al. [32]. Une explication est une interaction entre deux agents: l'utilisateur et l'explicateur. Cette interaction est souvent initiée par l'utilisateur avec une question à propos du phénomène à expliquer, d'où son influence sur celui-ci. L'explicateur formule une explication pour répondre à l'utilisateur. L'explication est composée du phénomène à expliquer, les causes de ce phénomène et le raisonnement qui permet de lier les causes au phénomène. C'est l'explicateur qui fabrique l'explication et donc qui choisit les causes et le raisonnement pour expliquer le phénomène à l'utilisateur. Miller [33] avance que l'explication dépend des croyances que l'explicateur a à propos de l'utilisateur. Par exemple, l'explicateur ne formulera pas la même explication s'il pense que l'utilisateur est un expert du domaine ou un non-initié.

Chaque utilisateur a des attentes différentes sur les causes et le type de raisonnement qui lie les causes au phénomène. Ces attentes dépendent des connaissances, expériences et croyances de l'utilisateur [34, 35]. Afin de simplifier la génération d'explications, les chercheurs proposent trois catégories d'utilisateur:

Experts en IA Ils sont intéressés par les explications techniques qui leur permettent de déboguer et d'améliorer le système d'IA expliqué. Ils ont une bonne compréhension du fonctionnement des systèmes d'IA et sont capables de comprendre des explications sophistiquées et techniques.

Experts du domaine Ils sont experts dans le domaine d'application du système d'IA expliqué. Ils cherchent à comprendre les causes d'une décision particulière et à évaluer la précision du système. Ils peuvent également avoir besoin d'explications détaillées pour informer d'autres personnes (par exemple des clients ou des patients).

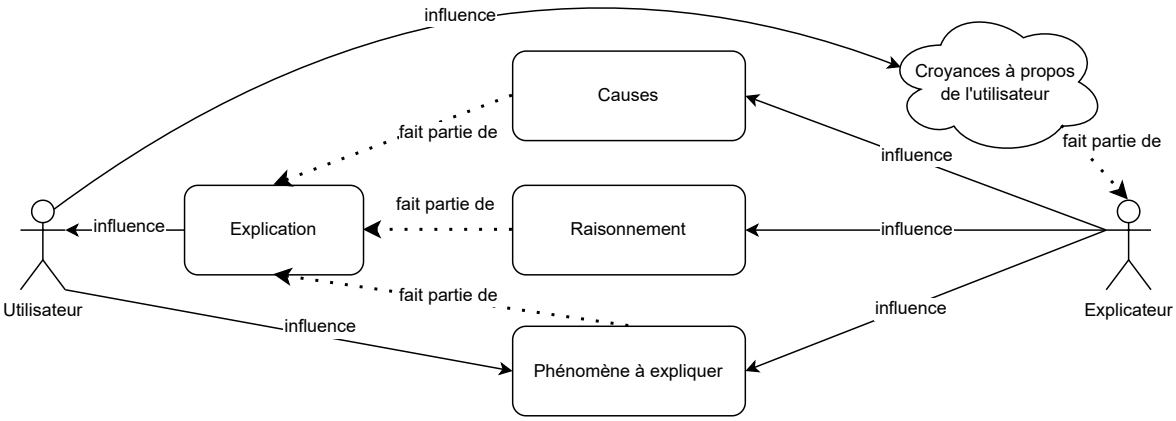


Figure 1.2: Diagramme d'une explication. Les flèches représentent une influence directe de la source sur un élément (par exemple l'explicateur choisit les causes). Les flèches en pointillés représentent une composante (par exemple les causes sont une composante de l'explication).

Non-initiés Ils n'ont pas d'expertise particulière en IA ou dans le domaine d'application du système d'IA. De la même manière que les experts du domaine, ils cherchent à comprendre les causes d'une décision particulière mais ont besoin d'explications simples et non techniques.

Parmi les choix à faire pour créer une explication adaptée à l'utilisateur, l'explicateur doit choisir un type de raisonnement. Les humains utilisent différentes formes de raisonnement.

Déduction La déduction est appliquée pour effectuer des démonstrations. Elle utilise la logique pour tirer une conclusion à partir d'un ensemble de prémisses. Une inférence déductive est toujours vraie si l'ensemble des prémisses est vérifié. Mais ce type de raisonnement n'est pas adapté pour expliquer un système d'IA aux experts du domaine ou aux non-initiés car elles nécessitent de représenter les causes et phénomène dans le vocabulaire de la logique du premier ordre [36]. De plus, les modèles d'apprentissage automatique suivent un raisonnement inductif qui empêche la construction d'une explication déductive.

Induction L'induction est un mode de raisonnement qui tire des conclusions en se fondant sur un ensemble d'observations. Contrairement à la déduction, la véracité des conclusions dérivées de ces observations n'est pas garantie. L'induction peut être décrite comme une "généralisation" [37]. Cependant, il faut une quantité inconnue d'observations pour pouvoir généraliser, ce qui n'est pas possible pour expliquer une décision particulière.

Abduction L'abduction est un raisonnement qui cherche à identifier la cause la plus probable d'un phénomène donné. Miller [33] décrit le processus d'abduction comme suit: (1) on observe un évènement; (2) on génère une ou plusieurs hypothèses à propos de l'évènement; (3) on juge la plausibilité de chaque hypothèse; (4) on sélectionne la "meilleure" hypothèse comme explication. Le raisonnement par abduction est vu comme particulièrement pertinent pour générer des explications par certains chercheurs [33, 37, 38]. Néanmoins, peu de travaux en XAI utilisent explicitement ce mode de raisonnement pour générer des explications.

Raisonnement contrefactuel Le raisonnement contrefactuel consiste à explorer l'influence de certaines causes sur le phénomène à expliquer. Ce raisonnement consiste à imaginer l'issue d'un évènement en modifiant l'une de ses causes. Par exemple, "je ne serais pas arrivé en retard si mon réveil avait sonné". Le phénomène à expliquer est le retard et une cause identifiée est le fait que le réveil n'ait pas sonné. Le raisonnement contrefactuel cherche à modifier les faits et explorer l'impact de ces modifications sur un phénomène. Cette forme de raisonnement est très populaire dans la communauté du XAI car elle est techniquement faisable et particulièrement pertinente pour générer des explications [39].

1.2.2 Évaluation d'une explication

L'évaluation de la qualité d'une explication est une tâche complexe du fait des attentes très variées des utilisateurs. Toutefois, la communauté du XAI explore différents critères qu'une explication doit remplir afin de garantir une explication de qualité. Ces critères sont divisés en trois catégories par Yang et al. [40]: la *généralisibilité*, la *fidélité* et la *capacité de persuasion*. D'autres chercheurs ont proposé différents critères qui peuvent être considérés comme une décomposition de ces trois critères. La *généralisibilité* est un indicateur de la capacité d'un utilisateur à généraliser le fonctionnement d'un système d'IA à partir d'un ensemble d'explications. Une explication généralisable permet de mieux anticiper le résultat d'un système d'IA dans des situations différentes de celle qui a été expliquée. La généralisibilité dépend de la *complétude* d'une explication [41, 42], c'est-à-dire que la majorité des causes de la décision ont été identifiées. De même, la *continuité* et l'*uniformité* des explications jouent un rôle dans sa capacité de généralisation [42]. Ces deux critères s'assurent que pour des situations proches, les explications sont similaires. Cela permet à l'utilisateur de comprendre et généraliser le fonctionnement d'un système d'IA pour des situations proches.

La *capacité de persuasion* d'une explication correspond à comment l'utilisateur comprend et réagit à l'explication. Elle mesure la satisfaction et la compréhension d'un groupe d'utilisateurs vis-à-vis d'une explication. Pour qu'une explication soit persuasive, différents critères comme la *clarté*, la *cohérence*, le *contexte* ou encore la *taille* de l'explication entrent en jeu [41, 42, 43]. La *clarté* s'assure que l'explication n'est pas ambiguë, la *cohérence* et le *contexte* correspondent au fait que l'explication est cohérente avec les croyances et expériences de l'utilisateur et les utilise pour identifier les causes. La *taille* d'une explication représente la quantité d'informations fournies. Le consensus veut que les explications ne soient pas trop longues car cela nuit à sa capacité de persuasion. Cependant, ce critère s'oppose au critère de *complétude* et donc de *généralisibilité* qui cherche au contraire, à fournir le maximum d'informations.

Enfin, la *fidélité* d'une explication décrit la capacité d'une explication à décrire correctement et précisément le processus de décision et présenter les causes réelles d'une décision [40, 41, 42]. Ce critère s'assure que les causes ne sont pas inventées mais sont fidèles au fonctionnement du système d'IA. C'est l'un des critères les plus importants d'une explication car une explication persuasive et généralisable mais fausse détériore la confiance de l'utilisateur en le système.

La mesure de chacun de ces critères est complexe et ne fait pas l'objet de consensus. La *fidélité* est la plus facile à mesurer de manière objective car elle n'a pas besoin d'intervention humaine. A l'inverse, la *capacité de persuasion* et la *généralisibilité* sont des critères subjectifs qui nécessitent donc des études utilisateurs pour les mesurer. C'est pourquoi des sous-critères objectifs comme la *taille* ou la *complétude* sont étudiés. Cela permet de fournir des mesures objectives pour des critères subjectifs et donc difficilement quantifiables. La création de mesures et de protocoles d'évaluation standardisés est activement recherché mais il n'existe pas encore de consensus.

1.2.3 Aperçu des méthodes d'explication

Dans cette section, nous présentons un aperçu des catégories de méthodes d'explications. Puis, nous discutons des méthodes d'explication dédiées aux images. Enfin, nous décrirons plus en détail les méthodes d'explications contrefactuelles.

Taxonomies du XAI

Malgré la nouveauté du XAI, de nombreuses méthodes d'explication ont été développées. La majorité de ces méthodes se concentrent sur expliquer les systèmes d'IA issus de l'apprentissage automatique. Afin de mieux comprendre les points communs et différences entre les différentes méthodes, plusieurs taxonomies des méthodes d'explicabilité ont été proposées [44]. Cependant, ces taxonomies ne font pas consensus. Cela est principalement dû au fait que la classification des méthodes de XAI dépend de l'objectif et de l'audience de celle-ci. Toutefois, on constate des

points communs dans la classification. Trois catégories de méthodes font consensus. La distinction entre les explications *globales* et *locales* correspondent au phénomène qui est expliqué. Les explications *globales* cherchent à expliquer le fonctionnement du système d'IA tandis que les explications *locales* expliquent une décision particulière prise par le système d'IA [29, 34, 44, 45]. Les méthodes pour générer les explications sont divisées en deux catégories: *post hoc* et *ante hoc*[1]. Les méthodes *post hoc* expliquent un modèle déjà entraîné [44] en utilisant une méthode auxiliaire pour générer l'explication [34]. Quant aux méthodes *ante hoc*, elles exploitent directement le fonctionnement du modèle pour en dériver une explication, grâce à sa nature transparente ou interprétable [44, 46]. Enfin, une distinction est faite entre les méthodes *spécifiques* et *agnostiques* par rapport au modèle. Les premières ne fonctionnent que pour un type de modèle alors que les dernières fonctionnent pour n'importe quel type de modèle. On remarque que les méthodes *ante hoc* sont souvent *spécifiques* au modèle car elles exploitent directement son fonctionnement. Au contraire, les méthodes *post hoc* sont majoritairement *agnostiques* par rapport au modèle car elles utilisent une méthode auxiliaire pour générer l'explication.

Certaines taxonomies classent les méthodes de XAI selon leur fonctionnement. Trois catégories différentes sont identifiées par Speith [44]: l'*importance des variables*, les *modèles de substitution* et les *exemples*. Les méthodes d'explication par *importance des variables* calculent un score d'importance pour chaque variable d'entrée afin de présenter à l'utilisateur l'impact que chaque variable a sur une prédiction [28]. Les *modèles de substitution* sont des méthodes qui approximent le comportement du modèle à expliquer à l'aide d'un modèle interprétable. C'est ce modèle de substitution interprétable qui est ensuite expliqué à l'aide de méthodes *ante hoc* [34, 44]. Finalement, les méthodes fondées sur les *exemples* fournissent des exemples représentatifs du fonctionnement du modèle comme explication [29, 44].

Classifieurs d'images explicables

Les modèles de vision par ordinateur pour la classification d'images sont utilisés dans des domaines critiques en termes de sécurité comme la médecine ou les voitures autonomes. C'est pourquoi il y a un besoin de modèles performants et explicables. L'approche la plus commune pour expliquer les classifieurs d'images sont les méthodes d'importance des variables [47, 48]. Ces méthodes permettent de visualiser l'importance de chaque pixel pour la prédiction d'une certaine classe. Un exemple d'explication d'image par importance des variables est donné dans la Figure 1.3. Les méthodes d'importance des variables les plus populaires du XAI comme LIME [49], SHAP [50] ou DeepLIFT [51] sont des méthodes post hoc et agnostiques du modèle qui peuvent donc être appliquées aux classifieurs d'images.

D'autres méthodes dédiées à expliquer la classification d'image ont été développées. Deux approches sont explorées, les méthodes fondées sur l'occlusion et celles fondées sur le gradient [47]. Les méthodes fondées sur l'occlusion modifient certaines zones de l'image pour étudier la différence de prédiction qui en résulte. Cette différence de prédiction permet d'extraire l'importance des pixels modifiés. Ces méthodes ont l'avantage d'être agnostiques du modèle car elles ne font que modifier l'entrée et mesurer la différence de la sortie du modèle. A l'inverse, les méthodes fondées sur le gradient sont spécifiques aux réseaux de neurones. Elles calculent l'importance des pixels en exploitant le gradient de la classe prédite. L'importance de chaque région est mesurée par la magnitude du gradient de chaque pixel [53].

Les méthodes d'explication par importance des variables sont controversées. En effet, certains chercheurs discutent du manque de fidélité de ces méthodes [52, 54]. Molnar [54] a remarqué que l'état actuel de ces explications n'est pas satisfaisant à cause de leur fragilité et leur manque de fiabilité, combinés à un manque d'outils d'évaluation qui permettent de mesurer leur fidélité. De plus, la plupart des méthodes d'explication utilisent des algorithmes opaques pour générer les explications, ce qui les exposent aux problèmes de robustesses et fiabilité dont souffrent ces algorithmes. C'est pourquoi des approches alternatives sont en train d'être explorées. Par exemple,

[1] Les explications ante hoc sont aussi appelées modèles *transparents* ou *interprétables* mais leur appellation est sujette à débat dans la littérature [44]

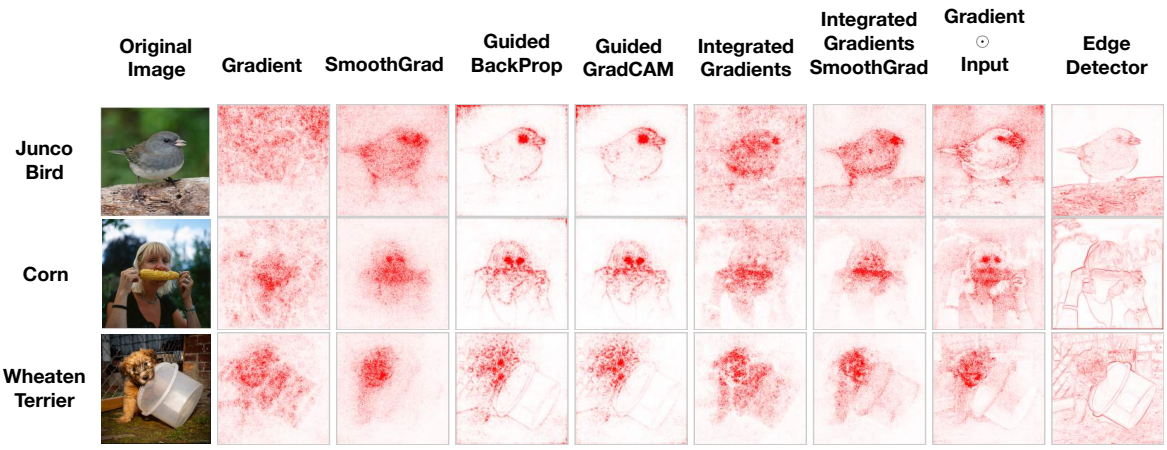


Figure 1.3: Exemple d'explication de classification d'image par importance des variables, par plusieurs méthodes populaires [52]

Pintelas et al. [55] a créé un XIS complet qui nécessite d'extraire des variables compréhensibles par les humains pour l'apprentissage et l'explication. L'intégration de connaissances experts sous forme d'ontologie pourrait être bénéfique à ces approches. Plusieurs systèmes neurosymboliques ont vu le jour, qui utilisent conjointement des ontologies et des modèles d'apprentissage automatique [56, 57, 58]. Cependant, la majorité de ces systèmes utilisent un algorithme boîte noire dans le processus de décision, ce qui limite l'explicabilité de ces systèmes. En outre, les systèmes neurosymboliques proposés ne sont pas conçus pour être explicable. Leur explicabilité est vue comme une conséquence de leur architecture, ce qui limite leur application pour résoudre les problèmes du XAI.

Explications contrefactuelles

Dans la Section 1.2, nous avons mentionné les explications contrefactuelles et leur récente popularité au sein du XAI. Une explication contrefactuelle est définie comme suit: *Une explication contrefactuelle pour une prédiction met en valeur les plus petits changements à faire sur les variables afin de modifier la prédiction vers un résultat prédéfini.* Ce type d'explication fait partie de la famille des méthodes d'explication par *exemples* car elles fournissent des exemples pour expliquer une décision. De nombreuses méthodes pour expliquer les décisions de modèles d'apprentissage automatique ont été développées. L'une des premières méthode est celle de Wachter et al. [39]. Ils définissent une contrefactuelle de la manière suivante.

Definition 1.2.1 (Contrefactuelle pour modèle d'apprentissage automatique)**.** Soit f un classifieur et $x \in \mathcal{X}$ un vecteur d'entrée tel quel $f(x) = y$, avec $y \in \mathcal{Y}$ la classe prédite. Une contrefactuelle est un vecteur $\hat{x} \in \mathcal{X}$ qui suit les contraintes suivantes

$$f(\hat{x}) = \hat{y} \tag{1.1}$$

où $\hat{y} \in \mathcal{Y}$ est la classe désirée, telle que $\hat{y} \neq y$.

$$\hat{x} = \underset{x' \in \mathcal{X}}{\arg\min}\, d\left(x, x'\right) \tag{1.2}$$

avec d une métrique de proximité qui mesure la différence entre l'entrée originale et une contrefactuelle.

Les contrefactuelles permettent à un utilisateur d'identifier la frontière de décision entre la classe originale y et la classe désirée $\hat{y}$ [59, 60]. La contrainte de minimalité dans l'Equation (1.2) s'assure que la solution la plus proche de l'entrée originale est donnée, ce qui correspond aux plus petits changements à faire sur les variables. Plusieurs propriétés désirables des contrefactuelles ont été formulées afin de guider la génération de ces explications [45, 59, 60].

Validité La validité s'assure que la contrefactuelle suit la définition, c'est-à-dire qu'elle vérifie l'Equation (1.1).

Parcimonie La parcimonie représente la quantité de variables modifiées. Elle encourage des explications courtes afin de correspondre au critère de *taille* décrit dans la Section 1.2.2. On souhaite minimiser le nombre de variables modifiées afin de rendre l'explication la plus claire et concise possible.

Proximité La proximité est une métrique qui calcule la distance ou la similarité entre l'entrée originale et une contrefactuelle. Elle représente la difficulté d'effectuer les modifications proposées par la contrefactuelle [59].

Plausibilité La plausibilité est une mesure qui permet de déterminer si une contrefactuelle est réaliste ou a un sens dans le monde réel. Par exemple, une contrefactuelle qui modifie l'âge de l'utilisateur de 25 ans à 150 ans n'est pas plausible car cette valeur n'est pas réaliste.

Diversité La diversité est une mesure analogue à la proximité. Elle correspond à la distance ou similarité entre deux contrefactuelles. Certaines méthodes d'explications contrefactuelles cherchent à fournir plusieurs contrefactuelles qui sont les plus diverses possibles afin de ne pas fournir des explications redondantes.

Les méthodes de génération d'explications contrefactuelles pour expliquer les modèles d'apprentisage automatique fonctionnent de la même manière. Elles résolvent un problème d'optimisation qui cherche à trouver la ou les contrefactuelles qui optimisent les propriétés décrites. Deux stratégies sont utilisées: la résolution à l'aide d'un solveur qui trouve les solutions exactes ou la résolution à l'aide d'une heuristique qui est plus efficace mais trouve des solutions sous-optimales. Les méthodes de génération de contrefactuelles se distinguent par le choix de métrique de proximité et la formulation des propriétés dans le problème d'optimisation.

L'état de l'art des contrefactuelles pour apprentissage automatique a plusieurs limitations. Bien qu'il y ait un consensus sur les propriétés désirables et la terminologie, la manière d'atteindre ces propriétés n'est pas claire. Ce problème se retrouve dans le choix de la métrique de proximité ou dans la mesure de la plausibilité qui nécessite des connaissances experts. De plus, l'évaluation de ces explications est souvent faite via des mesures objectives, qui ne prennent pas en compte l'avis des utilisateurs [61, 62]. Même pour les évaluations objectives, il est parfois difficile de comparer les méthodes entre elles car elles n'utilisent pas les mêmes mesures de proximité et certaines méthodes ne permettent pas de les modifier.

1.2.4 Conclusion

L'une des premières difficultés rencontrés dans la littérature du XAI est le manque d'uniformité des termes. Il n'existe pas de consensus sur la terminologie du XAI, ce qui a pour effet de ralentir la recherche dans ce domaine. Une autre conséquence de ce manque de consensus est que les chercheurs se basent sur leur propre intuition de ce qu'est une explication pour développer de nouvelles méthodes. Il y a donc une très grande variété de méthodes d'explication mais qui partagent toutes les mêmes approches et donc les mêmes défauts. L'un des problèmes majeurs est le manque de fidélité des méthodes par importance des variables ou modèle de substitution, qui représente une grande majorité des méthodes. De plus, l'état de l'art du XAI se concentre particulièrement sur l'explication des modèles d'apprentissage automatique. Le développement des systèmes neurosymboliques ne bénéficie pas de ces méthodes d'explication, malgré leur fort potentiel pour la création de systèmes explicables. Nous identifions donc une direction de recherche encore peu explorée qui est la création d'un XIS suivant l'architecture de la DARPA, qui se fonde sur un système neurosymbolique.

1.3 Contributions

Nos contributions s'articulent autour de la création d'un XIS qui utilise une ontologie. En premier lieu, nous proposons une terminologie du XAI afin de définir de manière non-ambiguë les termes utilisés dans cette thèse. Ensuite, nous présentons un XIS fondé sur les ontologies spécialisé dans la classification d'images. Enfin, nous développons une méthode dont le but est d'expliquer les prédictions de ce XIS. Cette méthode génère des explications contrefactuelles et est conçue pour être applicable à une majorité d'ontologies OWL.

1.3.1 Terminologie du XAI

L'intérêt récent pour la conception d'algorithmes d'IA explicables a conduit à la production d'un grand nombre de papiers de recherche dans ce domaine. Lors de notre étude de la littérature, nous avons observé un manque de consensus concernant la terminologie du XAI [28, 29, 63]. Dans cette contribution, nous étudions le vocabulaire employé dans la littérature afin d'identifier les termes et concepts récurrents du XAI et les possibles ambiguïtés liées aux noms et définitions de ces concepts. Nous résolvons ce problème en proposant une terminologie fondée sur nos observations de la littérature. Cette terminologie est divisée en deux parties. La première définit les termes liés à un système d'IA et la seconde définit les termes liés à une explication. Concernant la terminologie d'un système d'IA, nous définissions les termes fréquemment utilisés tels que *interprétabilité* ou *explicabilité*. Par la suite, nous identifions les relations entre ces termes. Les méthodes d'explicabilité sont perçues comme des systèmes d'IA particuliers qui partagent les mêmes propriétés tout en ayant un ensemble de caractéristiques propres comme la fidélité. Les relations entre les concepts décrivant un système d'IA que nous avons identifié sont formalisées dans une ontologie qui est alignée avec l'ontologie fondationnelle DUL [64]. L'objectif de cette ontologie est de faciliter la compréhension et l'adoption de notre terminologie.

La deuxième partie de la terminologie décrit la définition d'une explication. Nous définissons une explication comme étant le fruit d'une interaction entre l'utilisateur et l'explicateur. Les composantes d'une explications dans le contexte du XAI sont décrites et nous définissons également les notions d'explications *globales/locales* et *post hoc/ante hoc*. La plupart des notions liées à une explication font l'objet d'un consensus et ne sont donc pas explorées dans cette contribution. Enfin, nous positionnons notre définition d'une explication par rapport au patron d'ontologie pour définir une explication de Tiddi et al. [65]. Nous instancions ce patron pour représenter les explications en XAI, à l'aide de l'ontologie du système d'IA introduite dans la première partie de la terminologie. Nous décrivons notamment les causes de l'explication comme le résultat d'une méthode d'explicabilité et caractérisons l'influence qu'a l'utilisateur sur l'explication. En effet, bien que cette influence soit documentée dans la littérature, elle est absente du patron d'ontologie pour une explication. Nous notons que ce patron d'ontologie ne représente pas la nature interactive d'une explication mais dépeint plutôt l'explication résultante de ce processus interactif.

Dans l'ensemble, la terminologie proposée utilise des définitions générales qui permettent leur application à tout type de système d'IA et ne contredisent pas les définitions données dans d'autres papiers de recherche. Elle se concentre sur les termes récurrents et spécifiques du XAI qui sont définis de manière ambiguë dans la littérature, que ce soit en raison de leur nom ou de leur définition. Toutefois, cette terminologie présente certains inconvénients inhérents à cet exercice. Tout d'abord, elle n'est pas exhaustive, car il est pratiquement impossible d'identifier toutes les notions employées par le domaine du XAI et de les associer à des noms et des définitions adéquats. Deuxièmement, la terminologie et l'ontologie ne reflètent que notre compréhension et notre vision de ces termes dans le contexte du XAI. Une bonne terminologie est une terminologie qui est comprise et adoptée par l'ensemble de la communauté. Cette contribution n'est donc qu'une proposition ouverte au débat. Néanmoins, elle nous permet d'éviter les ambiguïtés dans cette thèse en définissant explicitement le vocabulaire utilisé. Dernièrement, nous avons observé une évolution rapide de la terminologie dans les deux années qui séparent la publication de notre terminologie (publiée en 2021) et la rédaction de ce manuscrit. La terminologie du XAI semble

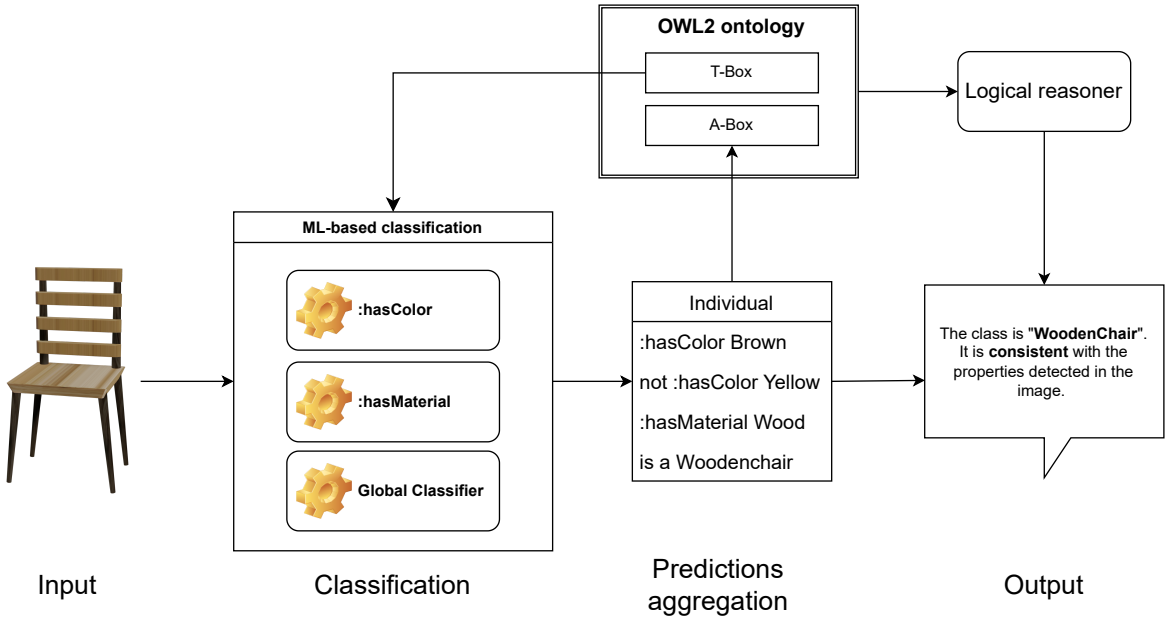


Figure 1.4: Diagramme du fonctionnement d'OBIC

converger vers une terminologie unique et partagée par la communauté. Dans le même temps, les travaux visant à identifier et mesurer les composantes de l'interprétabilité et l'explicabilité progressent et ajoutent donc de nouvelles notions à la terminologie du XAI. Ces notions peuvent remplacer ou affiner d'anciennes notions, dont certaines définies dans notre terminologie. Ainsi, notre proposition de terminologie est destinée à devenir obsolète dans les prochaines années; ce qui indique que le terminologie du XAI évolue vers une terminologie consensuelle.

1.3.2 Un système intelligent explicable fondé sur une ontologie pour classer les images

Une architecture de XIS a été produit par la DARPA [15] (voir Figure 1.1). Nous avons observé une absence de XIS suivant cette architecture et utilisant une approche neurosymbolique. Ainsi, nous créons un XIS composé d'un modèle explicable qui suit un processus d'apprentissage spécifique ainsi qu'une interface d'explication qui présente les résultats du modèle et les informations pertinentes et qui explique les prédictions. Le modèle explicable est spécialisé dans la classification d'image et est intitulé OBIC (pour Ontology-Based Image Classifier). OBIC utilise une ontologie pour créer de multiples modèles d'apprentissage automatique dont le rôle est de détecter des propriétés particulières d'un objet dans une image. Cette même ontologie est également utilisée comme un système de détection d'erreurs en vérifiant la cohérence des prédictions des classifieurs. La capacité du système à détecter et prévenir l'utilisateur lorsqu'une prédiction est incohérente est une étape vers des systèmes prédictifs de confiance et robustes. La conception d'OBIC est faite de manière à minimiser la quantité de travail requise pour l'implémenter. L'ontologie au coeur du système peut être une ontologie pré-existante qui nécessitera alors l'addition de propriétés d'objet et de définitions de classes pour être utilisable par OBIC. De même, OBIC est agnostique de l'architecture de modèle d'apprentissage automatique, ce qui permet la réutilisation de modèles existants et/ou l'utilisation de modèles explicables.

La Figure 1.4 illustre le fonctionnement d'OBIC. Tout d'abord, des modèles d'apprentissage automatique sont entraînés pour détecter une classe et la présence ou absence de concepts issus de la T-Box de l'ontologie. L'un des modèles, appelé le *classifieur global* est entraîné pour effectuer la tâche de classification principale, de la même manière que le ferait une architecture d'apprentissage automatique classique. Les autres modèles sont entraînés à détecter si des propriétés d'objet sont présentes dans l'image et, le cas échéant, déterminer la classe de l'objet. En-

suite, un individu représentant l'image est créé. La classe prédite par le *classifieur global* est attribuée à cet individu. De la même manière, les propriétés d'objet sont ajoutées à l'individu. Enfin, cet individu est inséré dans la A-Box de l'ontologie et un raisonneur logique est exécuté pour vérifier sa cohérence. La vérification de cohérence agit comme un système de détection d'erreur explicable car le raisonnement utilise la logique et des concepts humains, qui peuvent donc être expliqués.

Le dernier élément du XIS est l'interface d'explication. Celle-ci fournit les informations à propos du résultat du système afin d'accroître la transparence du XIS. Elle indique à l'utilisateur si la prédiction est cohérente et les raisons de la cohérence ou incohérence. En cas d'incohérence, les propriétés incohérentes sont indiquées afin d'aider l'utilisateur à comprendre le problème et décider s'il peut se fier à la prédiction finale. La disposition de l'interface fait en de sorte que les informations essentielles et compréhensibles par n'importe quel utilisateur (c'est-à-dire par les non-initiés, experts du domaine et experts en IA) soient visibles en premier. Les informations additionnelles sont également disponibles mais moins mises en avant, afin de permettre aux experts du domaine ou aux experts en IA d'avoir une meilleure compréhension de la décision prise par le système. Pour le moment, l'interface d'explication ne génère pas d'explication mais ne fait que présenter les informations du système. Afin de remédier à ce problème, nous proposons une nouvelle méthode d'explication décrite dans la Section 1.3.3.

Les performances du système de détection d'erreur d'OBIC ont été évaluées sur une tâche de classification d'instruments de musique. Une ontologie et un jeu de données ont spécialement été créés pour cette évaluation. L'expérimentation a révélé que OBIC est capable de détecter des incohérences dans les prédictions des classifieurs. Cette détection d'erreur est non supervisée car elle ne nécessite pas de connaître la bonne classe à prédire. A notre connaissance, peu de travaux ont été produits sur ce sujet. Par conséquent, ce système de détection d'erreur est un gain en explicabilité et fiabilité car OBIC a les mêmes performances que les modèles d'apprentissage automatique tout en étant capable de détecter des erreurs, ce qui ne peut être fait par les modèles classiques. Néanmoins, l'évaluation a révélé des points à améliorer sur la détection d'erreur. Le point principal est l'impossibilité de déterminer quel classifieur s'est trompé. Dans notre expérimentation, nous avons fait l'hypothèse que toute incohérence vient du *classifieur global*. Mais les résultats ont montré que cette hypothèse ne fonctionne pas car la probabilité qu'une incohérence vienne des modèles de propriété est proportionnelle au nombre de propriétés. Il y a généralement plus d'un modèle de propriété ce qui implique qu'il y a de plus grandes probabilités qu'une erreur provienne de ces modèles plutôt que du *classifieur global*.

1.3.3 Explications contrefactuelles pour les ontologies

Le XIS décrit dans la Section 1.3.2 utilise une ontologie pour détecter des incohérences dans ses prédictions. Cependant, nous ne disposons pas de méthode d'explication d'ontologie adaptée à tout type d'utilisateurs. Les méthodes d'explication d'ontologie sont destinées aux experts en ontologie et ne peuvent donc pas être utilisées. Les méthodes d'explication issues du XAI sont adaptées à tout type d'utilisateurs mais se concentrent sur expliquer les modèles d'apprentissage automatique. C'est pourquoi nous développons une nouvelle méthode d'explication pour ontologie qui est adaptée des méthodes du XAI pour apprentissage automatique. Étant donné le récent intérêt pour les explications contrefactuelles et leurs avantages décrits dans la Section 1.2, nous souhaitons créer une méthode d'explications contrefactuelles pour ontologies.

Dans un premier temps, nous adaptons le processus de génération de contrefactuelles pour modèles d'apprentissage automatique pour fonctionner avec les ontologies. Les méthodes de contrefactuelles pour apprentissage automatique utilisent un vecteur comme entrée et sortie. Nous redéfinissons ce vecteur afin de fonctionner avec les ontologies. Le vecteur étant un ensemble de variables, son équivalent pour les ontologies est un individu et ses assertions. Ainsi, nous définissons le graphe de connaissance d'un individu (IKG) comme étant l'ensemble des assertions dont l'individu est sujet. Une contrefactuelle pour ontologie est donc une version alternative de l'IKG original dont une ou plusieurs assertions ont été modifiées pour être cohérent

avec une nouvelle classe. Pour créer une contrefactuelle, il faut donc un IKG comme entrée et un nouvel ensemble de classes pour l'IKG. Par exemple, soit un IKG représentant une pizza qui a comme ingrédients de la viande hachée et des oignons. La classe de cette pizza est `PizzaAvec-Viande`. On cherche à déterminer les modifications à faire pour que la pizza soit végétarienne, c'est-à-dire que sa classe devienne `PizzaVegetarienne`. Sans modification sur ses assertions, ce nouvel IKG est incohérent car elle ne peut pas être végétarienne et contenir de la viande. Ainsi, les contrefactuelles sont des alternatives de cet IKG qui sont cohérentes avec le fait d'être une pizza végétarienne. L'une des contrefactuelles sera le même IKG auquel on a retiré l'ingrédient viande hachée.

La méthode que nous avons développé pour générer des contrefactuelles pour ontologies s'appelle CEO (pour Counterfactuals Explanations for Ontologies). Elle fonctionne sur le même principe que certaines méthodes équivalentes pour l'apprentissage automatique, ce qui signifie qu'une heuristique de recherche de contrefactuelles est utilisée. Nous définissons un espace de recherche qui contient l'entièreté des modifications possibles. Cet espace est représenté sous la forme d'un graphe d'édition. Chaque contrefactuelle est connectée aux autres par une succession d'opérations qui sont la suppression, l'insertion ou la modification d'assertions. L'heuristique de recherche explore ce graphe de manière à identifier les contrefactuelles cohérentes les plus proches possibles de l'IKG original. Pour ce faire, nous étudions les mesures de similarité entre individus d'une ontologie pour déterminer une métrique de proximité adéquate. Les propriétés désirables des contrefactuelles vues dans la Section 1.2.3 sont également applicables à notre méthode. La validité correspond à la cohérence de la contrefactuelle après vérification par un raisonneur logique. La proximité mesure la similarité entre la contrefactuelle et l'IKG original. La parcimonie correspond au nombre de modifications faites pour passer de l'IKG original à la contrefactuelle. Une fois l'espace de recherche exploré, nous filtrons les contrefactuelles pour ne garder que celles qui sont valides. Ensuite, nous calculons la proximité et la parcimonie de chacune pour identifier les meilleures explications à présenter à l'utilisateur.

La méthode a été évaluée sur deux expériences. Une première expérience avec l'ontologie Pizza[2] a pour but de valider la méthode et mesurer son temps d'exécution. Elle a permis de déterminer que la méthode CEO génère de bonnes contrefactuelles qui correspondent à nos attentes et objectifs. Cependant, l'expérience a montré une complexité algorithmique élevée qui rend cette méthode inapplicable à de grandes ontologies dans son état actuel. En effet, l'heuristique utilisée favorise une exploration en profondeur de l'espace de recherche plutôt que de minimiser le nombre de contrefactuelles explorées. Ensuite, une étude utilisateur a été effectuée pour évaluer la qualité et pertinence des explications contrefactuelles pour expliquer OBIC. Un objectif additionnel était d'évaluer la qualité de la mesure de proximité pour trouver les meilleures contrefactuelles. Cette étude utilisateur utilise la même tâche que l'évaluation d'OBIC, c'est-à-dire la classification d'instruments de musique dans une image. Six experts du domaine ont été interrogés. Neuf prédictions correspondant à neuf images différentes ont été présentées, avec les prédictions d'OBIC. Pour chaque prédiction, les dix meilleures contrefactuelles en terme de proximité et parcimonie leur ont été données. Les experts devaient cocher les explications qu'ils trouvaient pertinentes et sélectionner une explication préférée. Les résultats de cette étude montre que les utilisateurs trouvent cette manière d'expliquer pertinente et compréhensible. La majorité des explications ont été perçues comme pertinentes par les experts, ce qui indique que la mesure de proximité identifie correctement les explications pertinentes. Néanmoins, les explications préférées sont très rarement celles avec la plus faible proximité. De plus, les experts se sont plaints de la présentation et du nombre trop important d'explications. Il reste donc des points à améliorer concernant la mesure de proximité et la présentation des explications.

[2] `https://protege.stanford.edu/ontologies/pizza/pizza.owl`

1.4 Conclusions et travaux futurs

Dans cette section, nous résumons et discutons les contributions présentées dans cette thèse. Ensuite, nous présentons les perspectives d'amélioration et de développement de notre travail.

1.4.1 Conclusion

L'objectif principal de cette thèse était d'aborder un problème du XAI, à savoir la conception de méthodes d'explicabilité. Pour ce faire, nous avons utilisé des approches symboliques qui ont été identifiées comme une direction prometteuse pour créer des systèmes d'IA explicables. Notamment, les ontologies sont généralement considérées comme des candidats idéaux à cette fin car elles sont capables de représenter des notions utilisées par les êtres humains, sont lisibles par les machines et sont construites à l'aide de logiques de description. Nous avons effectué une revue de la littérature du XAI et identifié plusieurs problèmes ouverts. Le premier problème qui est apparu est l'absence de consensus concernant le vocabulaire du XAI. Ensuite, nous avons observé que le domaine de l'IA neurosymbolique, qui cherche à combiner les approches de l'IA symbolique avec l'apprentissage automatique, est très peu développé et n'explore pas le potentiel explicatif de ces nouveaux systèmes d'IA. Par conséquent, nous avons proposé la conception d'un système intelligent explicable tel que décrit par la DARPA [15] qui est centré sur une ontologie. Puis, nous avons exploré des méthodes pour expliquer ce XIS et développé une technique pour générer des explications contrefactuelles pour ontologies. Finalement, nous avons évalué le XIS et les explications contrefactuelles sur une tâche de classification d'images d'instruments de musique.

Notre première contribution, développée dans le Chapitre 4, concerne la terminologie du XAI. Nous avons identifié les termes importants du XAI et les avons définis en fonction de leur utilisation et de leurs définitions dans la littérature. La terminologie est centrée sur l'utilisateur car les explications sont spécifiques à chaque utilisateur. Nous avons créé une ontologie qui représente les concepts définis dans cette terminologie ainsi que leurs relations. Cette ontologie peut être utilisée pour catégoriser les systèmes d'IA. De même, nous avons fourni un patron d'ontologie pour définir les explications en XAI, fondé sur un patron d'ontologie pour définir des explications dans n'importe quel domaine. Bien que la terminologie soit fondée sur les définitions observées dans la littérature, elle ne fait que refléter notre compréhension du vocabulaire du XAI. D'autres discussions au sein de la communauté doivent être menées afin de parvenir à un consensus, ce qui pourrait prendre plusieurs années. Néanmoins, cette terminologie a permis d'éliminer les ambiguïtés qui aurait pu se répercuter dans nos contributions.

À la suite de cette terminologie, nous avons présenté la conception d'un XIS pour la classification d'image dans le Chapitre 5. Ce XIS exploite une ontologie pour construire le modèle et expliquer les prédictions. Le modèle explicable, OBIC, construit plusieurs modèles qui sont capables de détecter des propriétés observables définies dans l'ontologie. Les propriétés observables sont des propriétés de concepts qui peuvent être détectés dans les données et sont utilisées dans la définition de ces concepts. Ensuite, un système de détection d'erreur est appliqué, qui extrait les prédictions de chaque modèle et teste leur cohérence au regard de l'ontologie. Cette architecture permet l'explication du résultat de ce système en fournissant les propriétés détectées qui ont menées à la prédiction finale. De plus, le système de détection d'erreur constitue un outil qui aide l'utilisateur à décider s'il peut avoir confiance en la prédiction. Les prédictions et explications sont présentées grâce à un prototype d'interface d'explication qui affiche toutes les informations disponibles à propos de la prédiction et du système de détection d'erreur. Le système de détection d'erreur non-supervisé est évalué dans le Chapitre 7. Cette évaluation a montré qu'exploiter les incohérences de l'ontologie pour détecter les erreurs est une direction de recherche prometteuse.

La dernière contribution remédie directement au problème du manque d'explications pour OBIC. Elle introduit la méthode CEO dans le Chapitre 6 qui permet la générations d'explications contrefactuelles pour les ontologies. Son objectif principal est d'expliquer le résultat d'un raisonneur logique, tel que l'inférence de nouvelles assertions ou la détection d'une incohérence. Elle fonctionne en explorant un graphe de contrefactuelles qui sont des individus de l'ontologie et

en identifiant les explications les plus similaires à l'individu original. Elle est conçue pour être applicable à la majorité des ontologies comme outil pour déboguer une ontologie et expliquer les conclusions du raisonneur logique à des non-initiés. La méthode CEO a été évaluée dans le Chapitre 7, avec le même cadre expérimental que l'évaluation d'OBIC. Une étude utilisateur a été menée avec des experts du domaine pour déterminer la qualité et la pertinence des explications. Les objectifs de cette contribution sont atteints car l'évaluation a montré que la méthode est bien capable de déboguer une ontologie et d'expliquer le résultat d'OBIC aux experts du domaine.

Les systèmes d'IA qui utilisent des modèles d'apprentissage automatique pour prendre des décisions n'ont pas connaissances des concepts humains. Lorsqu'un modèle est entraîné, il apprend de lui-même un ensemble de concepts qui sont les plus appropriés pour mener à bien sa tâche. Cet ensemble de concepts est rarement aligné avec les concepts humains, ce qui rend le processus de décision impossible à comprendre pour les humains. Les méthodes d'explicabilité post hoc cherchent à identifier une correspondance entre les concepts du modèle et les concepts humains. Mais le résultat de ces méthodes n'est généralement pas fidèle et ne peut être appliqué pour des applications sensibles. En revanche, les ontologies peuvent être combinées avec des modèles d'apprentissage automatique afin de s'assurer que le modèle utilise les concepts humains extraits de l'ontologie durant l'apprentissage. Ainsi, les concepts utilisés par le modèle sont déjà connus et compris des humains, ce qui facilite la création d'explications fidèles. En outre, les ontologies utilisent un raisonnement déductif pour inférer de nouveaux faits, fondés sur les connaissances préalables. Ce mode de raisonnement peut aisément être expliqué en retraçant les prémisses de l'inférence. À l'inverse, les modèles d'apprentissage automatique utilisent un raisonnement inductif qui ne peut être facilement expliqué car il ne suit pas de processus logique. Les humains utilisent à la fois les raisonnements inductif et déductif pour faire des inférences. C'est pourquoi combiner les ontologies et les modèles d'apprentissage automatique est idéal pour répliquer le raisonnement humain et donc établir un processus de décision plus compréhensible par les humains. OBIC a été conçu dans cette optique de combiner les raisonnements inductif et déductif. Par exemple, une personne peut expliquer pourquoi elle a vu une chaise en bois, en décrivant la présence de concepts tels que des pieds de chaise, un dossier ou des accoudoirs ainsi que la texture du bois ou une couleur spécifique au bois. Tous ces concepts assemblés en un seul objet résultent en une chaise en bois, en appliquant un raisonnement déductif fondé sur les connaissances de la personne. Toutefois, lorsqu'une personne est questionnée sur pourquoi elle a vu une certaine couleur, cette personne ne peut fournir ce même raisonnement déductif. À la place, elle utilise probablement un raisonnement inductif fondé sur ses expériences, qui ne peuvent être aisément expliquer à une autre personne. Chaque humain a un processus de décision unique pour identifier ces concepts basiques comme la couleur ou le son. OBIC suit le même fonctionnement en utilisant un raisonnement inductif pour détecter les concepts basiques, grâce aux modèles d'apprentissage automatique, puis applique un raisonnement déductif à partir de connaissances humaines pour inférer un fait, à partir des concepts détectés. En résumé, la combinaison d'approches symboliques et de modèles d'apprentissage automatique a le potentiel de répliquer le raisonnement humain et d'exploiter les connaissances humaines pour prendre des décisions précises et explicables. Cette combinaison est en cours d'exploration par le domaine de l'IA neurosymbolique. Dans cette thèse, nous avons proposé une méthode neurosymbolique dédiée à l'IA explicable. Nos contributions atteignent notre objectif de concevoir un système intelligent explicable qui exploite l'IA symbolique. Dans la section suivante, nous discutons des perspectives pour améliorer ces contributions.

1.4.2 Travaux futurs

Lors du développement de notre XIS, nous n'avons pas identifié de méthodologie afin de l'adapter à une tâche spécifique, une certaine audience et un domaine d'application. À notre connaissance, très peu de travaux ont été faits sur cette problématique. La terminologie et l'ontologie des systèmes d'IA pourraient être appliquées pour développer une méthodologie qui permet de choisir les modèles et les techniques de XAI adéquats pour créer un XIS. La terminologie a besoin d'être

mise à jour afin de prendre en compte les les dernières avancées dans le domaine. De même, elle pourrait être étendue pour inclure une partie au sujet de l'évaluation des explications et des méthodes de XAI. Cette extension pourrait également être utilisée pour affiner la méthodologie de construction d'un XIS. Une potentielle direction de recherche pour concevoir une telle méthodologie est de choisir une tâche réelle et collaborer avec l'audience cible pour construire un XIS qui répond à leurs besoins. En effet, le problème principal de l'évaluation d'OBIC et de CEO est que la tâche ne correspond pas à un besoin réel et le développement du XIS n'a pas été fait en coopération avec les utilisateurs ciblés. Appliquer OBIC et CEO à une tâche avec des besoins clairement identifiés pourrait aider à faire des choix quant aux paramètres d'OBIC, la métrique de proximité pour CEO ou l'implémentation de techniques d'explicabilité additionnelles afin d'expliquer le résultat d'OBIC.

Les expérimentations sur OBIC ont montré que les performances des classifieurs affectent le système de détection d'erreur. Une méthode pour améliorer les performances des classifieurs pourrait être d'exploiter l'interface d'explication pour que les humains puissent étiqueter les données inconnues et donc obtenir plus de données d'apprentissage. De plus, un système capable d'identifier le classifieur responsable de l'incohérence pourrait être ajouté. Lors de l'évaluation d'OBIC, nous avons observé que les explications fournies par l'interface d'explication ne sont pas fidèles, car elles sous-entendent que les classifieurs de propriété influencent la classification, or ce n'est pas le cas. Ainsi, l'architecture actuelle où le *classifieur global* est le seul classifieur responsable de la classification principale doit être modifiée pour rendre les explications plus fidèles au système. Une manière de remédier à ce problème est de retirer le *classifieur global* et déterminer une méthode déductive pour effectuer la classification. Cela pourrait être accompli en utilisant l'ontologie pour trouver une liste de classes qui sont compatibles avec les propriétés détectées. Les classes auraient alors besoin d'être triées afin de prédire la classe finale. La méthode CEO pourrait être appliquée pour explorer ce qu'il adviendrait de la classe finale lorsque d'autres propriétés sont détectées.

Toujours à propos d'OBIC, nous pensons que ce système pourrait être une instance d'une architecture plus générique. Cette architecture générique peut être décrite selon ces deux étapes: détecter des concepts compréhensibles par les humains à partir des données brutes, puis appliquer un raisonnement logique sur ces concepts pour prendre ou confirmer une décision. Une transformation des données brutes en concepts intermédiaires humains se fait grâce à des algorithmes inductifs (par exemple des modèles d'apprentissage automatique). Ces concepts humains sont déterminés par les connaissances expert sous la forme d'une ontologie. Ensuite, un raisonnement déductif est appliqué pour prendre la décision finale, fondée sur ces concepts. OBIC est une instance de cette architecture, où les données brutes sont des pixels, les concepts intermédiaires sont des textures, des formes ou des concepts plus élaborés comme les mécanismes d'un instrument de musique. Puis, un raisonneur logique utilise ces concepts pour vérifier la décision. D'autres études sur cette architecture générique doivent être menées. De même, des instanciations de cette architecture sur d'autres tâches avec d'autres types de données doivent être explorées.

Concernant CEO, les perspectives principales sont l'extension des types d'assertion gérés, l'addition des opérations d'insertion dans la recherche heuristique et l'amélioration de la complexité algorithmique. L'ajout de nouvelles assertions et des opérations d'insertion aura pour effet de détériorer la complexité algorithmique. Il apparaît donc qu'il faut une nouvelle méthode pour explorer les contrefactuelles qui permet à l'utilisateur de choisir entre un temps d'exécution plus long pour plus de diversité ou un temps d'exécution plus court au risque d'avoir des explications moins pertinentes. À propos de la diversité, des travaux futurs doivent être conduits pour ajouter cette propriété au classement des contrefactuelles afin de résoudre certaines limitations observées durant l'évaluation de CEO. Quant au classement des contrefactuelles, l'évaluation a montré l'inadéquation de la mesure de proximité utilisée. Le choix de la métrique de similarité pour les classes de l'ontologie est la cause de ce problème. De nombreuses métriques de similarité existent dans la littérature afin de mesurer la similarité entre classes, des métriques qui seront

explorées afin de résoudre notre problème de choix de proximité.

Finalement, nous notons que la méthode CEO a ouvert de nouvelles opportunités pour les contrefactuelles pour apprentissage automatique. Notre revue de la littérature a exposé plusieurs problèmes dans les méthodes qui génèrent les contrefactuelles pour expliquer ces modèles. Notamment, deux problèmes principaux apparaissent, la conception d'une métrique de proximité entre les variables qualitatives et l'identification de critères de plausibilité. Nous pensons que le type de métrique de proximité utilisée pour CEO pourrait être appliqué dans le contexte de l'apprentissage automatique pour mesurer la similarité entre les variables qualitatives. De manière similaire, l'exploitation d'ontologie pour déterminer les critères de plausibilité semblent être une direction intéressante à explorer.

Chapter 2

Introduction

The subject of this thesis lies at the intersection of different fields, but mainly focuses on eXplainable AI (XAI). We introduce the context of this thesis in Section 2.1. Afterwards, we introduce the domain of XAI in Section 2.2 to give an overview of its goals and current issues. Finally, we discuss the motivations for this thesis and present the structure of this manuscript in Section 2.3.

2.1 Context

Modern Artificial Intelligence (AI) appeared in the late 1940s thanks to the recent advancements in automated computation and formal logic. In its history, the field of AI saw three "summers" and two "winters" that designate periods where funding for research in AI were either abundant or scarce. We are currently in the third summer of AI that is driven by machine learning and more specifically deep learning [1]. Indeed, from computer vision to natural language processing, there is not a single domain that has not been affected by breakthroughs in neural networks. This led to a wide adoption by industries in multiple domains such as healthcare, justice, automobile industry, or even art [2]. Artificial neural networks are flexible models that can approximate mathematical functions, given the right parameters. These parameters are commonly found with the back-propagation algorithm that uses gradient descent to update them for every new observed data point. However, this is computationally expensive and requires a large amount of data, especially considering the size of recent neural networks such as GPT-3 that have hundreds of billions of parameters [3]. Hence, the potential of neural networks was unlocked by the recent increase in computing power and the availability of large amounts of data, leading to this third summer of AI that started around 2012 [1]. Nevertheless, neural networks suffer from several drawbacks that make their application in certain fields undesirable. One of those limitations is the inability to explain the decision made by a neural network. Therefore, the application of neural networks is frowned upon in critical domains that have a direct impact on human lives. This explainability issue was also met in the past AI summer that saw the rise of expert systems. Still, expert systems are symbolic AI algorithms that have the advantage of using human knowledge and deductive reasoning for their decision process. This property of expert systems motivated scholars to explore their combination with neural networks to solve the explainability problem.

Expert systems were as popular in the 1970s as neural networks are today. They come from advances in symbolic AI, a class of AI algorithms that manipulate symbols that may be human understandable. Like neural networks, expert systems have rapidly been used then in many domains for a variety of tasks. It is estimated that over two thirds of the Fortune 1000 companies were involved in applying expert systems in daily business activities in the 1980s [4]. Expert systems and generally knowledge-based systems apply formal logic on human knowledge to make a decision or a prediction. Two main components interact to solve a task: a knowledge base and an inference engine [5]. The knowledge base contains the specific domain knowledge necessary for solving problems. It uses some form of representation such as rules or semantic networks. This knowledge is acquired with the help of domain experts that collaborate with a an engineer to encode

their knowledge and experience. Then, the inference engine is capable of reasoning and interpreting the rules and knowledge of the knowledge base. Its task is to find logical paths in the forest of rules to arrive at a conclusion. The sudden interest in expert systems was due to their ability to automatically replicate the decision process of a human expert. It enabled companies to save time and money on repetitive but highly specific tasks. The first example of a successful adoption of an expert system in the industry is with XCON [6] that dramatically decreased the process of configuring a custom computer from 90 days to 90 minutes [1].

The common denominator between machine learning and symbolic AI is their objective to replicate human reasoning. Indeed, expert systems and neural networks mimic a subset of human intelligence: the ability to reason and deduce from facts for expert systems and the ability to learn and induce new knowledge for neural networks. However, as mentioned earlier, a key human behavior is the ability to explain, especially for high stakes decisions. According to several scholars that worked on expert systems, an AI system must be able explain its decision process to ensure user acceptance [7]. Thanks to the symbolic nature of expert systems, explanations are easy to generate and usually consists in tracing the logical steps that led to a decision. Conversely, explaining neural networks is a significantly harder task since they do not use human-understandable symbols. Moreover, the functioning of neural networks is much more complex to understand and require advanced knowledge in mathematics to understand both the training and inference phases. This lack of explainability may lead to disastrous consequences if their decisions were blindly trusted. Indeed, when learning from biased data, deep learning models will reinforce these biases and consequently make discriminatory decisions [8]. Motivated by these issues along with the application of the GDPR regulations by the European Union that promote a "right to explanation" [9], the research area of eXplainable AI is rapidly gaining popularity and explored by academics and industries.

In addition to the problem of explainability, neural networks have other flaws. Namely a lack of robustness and the need for massive amounts of data and computation power [10]. In their time, expert systems also suffered from intrinsic flaws such as their inability to handle reasoning with uncertainty and the difficulty to acquire sufficient expert knowledge [1]. These issues led to the demise of expert systems and the beginning of an AI winter. Neural networks may face the same fate if no solution is found. Fortunately, scholars have recently advocated for the hybridization of symbolic AI and neural networks to overcome the flaws of both approaches. This idea is also motivated by the fact that human intelligence is capable of both deduction and induction. In order to replicate human intelligence, it seems sensible that AI systems should also be able to do both. The field of research that seeks to create hybrid AI methods that combine neural networks and symbolic AI is named neurosymbolic (or neural-symbolic) AI.

In the history of AI, the winters were caused by the inability to remedy the limitations of the popular methods. The AI community fears a third AI winter that would be provoked by the lack of solutions to the aforementioned issues of deep learning. The fields of explainable AI and neurosymbolic AI are new domains that seek to address these limitations and avoid the fate of the last two AI summers. The relative youth of these domains means that many research directions are yet to be explored. Consequently, these fields are not standardized, especially in XAI where fundamental notions are not clearly defined in a consensual way. In addition, the XAI domain is confronted to difficulties outside the scope of computer science. Particularly, the definition and evaluation of an explanation is a multidisciplinary problem, including social sciences, psychology or even philosophy. Meanwhile, research in XAI is growing quickly. A large number of methods have already been created but still no standardized way to evaluate and compare them. Finally, XAI and neurosymbolic AI seem to follow distinct paths with different objectives. Explainability is usually mentioned as a possible application of neurosymbolic AI methods, but it is rarely explored and evaluated. As the name of this thesis suggests, it aims at exploring the use of symbolic and neurosymbolic methods to address current challenges of the XAI field.

2.2 Introduction to Explainable AI

Explainable AI is a field that was created in response to the general need for explainability in AI along with the manifest opacity of current machine learning models. Providing explanations that are satisfying to every user is challenging since users have different expectations, beliefs, knowledge and needs. The design of an explanation not only depends on the explainee but also on the explainer's objectives and the overall context [11]. For instance, a physician does not explain a diagnosis in the same manner, whether they are talking to a patient, a student or a colleague. They seek to earn the trust and acceptation of the patient, pass on knowledge to the student and justify the diagnosis to the colleague. Each explanation will therefore be built differently to achieve each goal. Since XAI produces methods to generate explanations, it is crucial to have a grasp of the different goals XAI aims at achieving to better understand the landscape of XAI and its current issues.

Two main goals of an explanation were identified by scholars: help the user understand a system and build trust with it [12, 13, 14]. The DARPA research program on XAI [15] is often referred as the starting point of modern XAI and used as a reference. The goal of this program is *to create a suite of new or modified ML techniques that produce explainable models that, when combined with effective explanation techniques, enable end users to understand, appropriately trust, and effectively manage the emerging generation of AI systems* [16]. Trust is thus seen as a fundamental goal of XAI. Moreover, recent developments in AI have raised new problems that may be addressed by XAI. Very recently, new conversational agents have been created and made available to the public. Particularly, Meta (the company that owns social media such as Facebook or Instagram) launched Galactica, a language model trained on scientific articles, capable of storing, combining and reasoning about scientific knowledge [17]. Although it showed promising results based on standard benchmarks, users found that Galactica only generated fake information in a confident way [18]. The public was quick to denounce this AI as unethical and even dangerous. In the same period, a similar model was made available: ChatGPT. In contrast to Galactica, ChatGPT was met with an incredible success and popularity reaching millions of daily users in a few days [19]. The great performance of this model led to questionable use, e.g. several research papers were published with ChatGPT as a co-author [20]. Consequently, several ethical questions concerning AI and its use were raised and actions were taken to prevent abusive uses. Nevertheless, the creators of ChatGPT had already implemented a safety system to prevent these issues, which detects and filters undesired content [21]. Despite these efforts, users managed to find ways to bypass this safety system to generate harmful content, which exposed the lack of robustness of deep learning models. Finally, in addition to the ethical questions raised by the use of AI, ChatGPT also raised such questions about the development of AI. Indeed, the news magazine TIME revealed that in order to train the safety system, workers were hired to manually annotate undesirable data exposing them to extreme content [22]. With these recent developments it is clear that in order for AI models to be trusted, explainability and interpretability is not sufficient. Other requirements must be met to ensure that AI systems are ethical and responsible both in their development and their utilization.

Conscious of these growing problems and their nefarious consequences, several entities ranging from private companies to governments have proposed guidelines or principles for the development and use of a responsible, trustworthy or ethical AI [23, 24, 25]. Fjeld et al. [26] reviewed 36 documents that propose guidelines for the design of a Responsible AI, coming from civil society, governments, inter-governmental organizations, multi-stakeholders and the private sector. They identified trends and themes that are mentioned in the majority of the 36 documents. Lately, documents have converged towards key principles that constitutes the "normative core" of Responsible AI. In a recent review of responsible AI, Mikalef et al. [27] propose the following principles and their respective descriptions:

Fairness AI systems should enable inclusion and diversity and not lead to discriminatory outcomes.

Transparency AI systems should be open and transparent regarding processes and outcomes and

facilitate traceability, explainability and user communication.

Accountability AI systems should be developed considering the responsibility and accountability of their outcomes with ethics and principles.

Robustness and safety AI systems should be developed with a preventative approach to risks and in a manner that they behave as intended while minimizing unintentional and unexpected harm.

Data governance AI systems should ensure that adequate data governance covers the quality and integrity of the data throughout the entire life-cycle.

Laws and regulations AI systems should adhere to the respective laws and regulations that dictate their functioning.

Human oversight AI systems should generate tangible benefits for people and always stay under human control.

Societal and environmental well-being AI systems should promote ecological and social responsibility, sustainability and not cause any harm.

Although XAI seems to be a solution exclusively for transparency, it can also be applied to address other principles. Among the 32 different principles identified by Fjeld et al. [26], 28 principles explicitly include XAI as a crucial component according to Arrieta et al. [28]. Indeed, explanations can be used to achieve different goals. Understanding what features were used to make a prediction may indicate whether the AI system is discriminatory or biased, a requirement for a fair AI. Likewise, explanations facilitate the audit of an AI system and the report of negative impacts, which is needed to ensure accountability. This new ideal of a responsible AI uses XAI as the principal means to achieve its goals. Therefore, XAI's goals have expanded and new research directions have been identified. In the meantime, other goals for XAI have been discussed by Adadi and Berrada [29], that overlap with the objectives of responsible AI. Namely, explain to justify, explain to control, explain to improve and explain to discover. Explanations to justify and control almost entirely encompass what was previously described. Indeed, justification ensures fairness and auditability which leads to building trust while explaining to control is about detecting and preventing errors and system failures, therefore ensuring robustness and safety. Explanations to improve and discover do not directly contribute to responsible AI or even to building trust, but they may lead to new significant progress in AI if pursued. According to Adadi and Berada [29], a model that can be explained and understood is one that can be easily improved. They argue that XAI could be the foundation for ongoing iteration and improvement between human and machine. While explaining to improve deals with how humans can improve machines, explain to discover is the other way around. Machine learning models learn from data; thus they might discover new knowledge, observe new correlations that are unknown to mankind. For example, AI systems based on reinforcement learning now excel in games like chess or game of Go. It would be desirable to understand the learned strategies, as a way to increase human knowledge. More generally, with the application of learning algorithms to different scientific fields, discovering the knowledge learned by these models could lead to scientific breakthroughs.

According to DARPA's XAI program, research in XAI can be organized into three areas [30]:

1. Understanding the psychology of explanation by summarizing, extending and applying psychological theories of explanation.

2. The development of new XAI methods for machine learning and explanation techniques for generating effective explanations.

3. Evaluation of the new XAI techniques in two challenge problem areas: data analytics and autonomy.

These three problems are representative of the current challenges of XAI. In this thesis, we explore solutions to these challenges by using symbolic approaches, especially Semantic Web Technologies that have been identified as a promising in the literature.

2.3 Motivations and outline

This manuscript introduces our contributions that are geared towards exploring new solutions to solve the explainability problem. In this section, we discuss our motivations for this thesis. Afterwards, we present the organization of the manuscript.

2.3.1 Motivations

The main issue that transpires in every aspect of XAI is a general lack of consensus. Indeed, as reflected by the challenges coined in the previous section, the definition and evaluation of an explanation are still being debated. As a result, the terminology of XAI is the first aspect to suffer from the lack of consensus. We observed identical terms that have different definitions and different terms that have similar or overlapping definitions. For instance, the terms explainability and interpretability are sometimes defined as synonyms and other times defined differently although they are among the most important terms in XAI. Moreover, authors do not systematically define terms used in their papers, thus increasing the general confusion in the terminology. This problem has consequences on the definition, design and evaluation of explanations while rendering the XAI field particularly difficult to understand to newcomers.

In relation to the issue of terminology, the identification of relevant criteria to evaluate an explanation and XAI methods is being discussed. Scholars seem to share the same view on the criteria that represent the explanation quality. Still, there is no consensus on the names, definitions and corresponding mathematical formulae for these criteria. Moreover, explanations are a social process that imply a part of subjectivity in the evaluation. Yet, the evaluation of XAI methods is mostly limited to objective metrics. Several literature reviews on the evaluation of XAI methods noted the scarcity of human subject evaluations, mostly due to their cost and difficulty to setup. The few existing user-studies have confirmed that the quality of an explanation is dependent on the user and context.

Nevertheless, the community agrees on the division of XAI methods into two categories: *post hoc* and *ante hoc*. *Post hoc* methods are ideal to explain black-box algorithms while *ante hoc* methods exploit the interpretable nature of some AI algorithms to generate an explanation. Several authors are advocating for the use of *ante hoc* methods and therefore interpretable models. They argue that *post hoc* methods lack robustness and faithfulness which could lead to counterproductive results. Conversely, "traditional" interpretable models (e.g. linear models, decision trees or rules) generally achieve poorer performance than their opaque counterparts, especially when handling high dimensional data. Even in cases where interpretable models match the performance of opaque models, the high dimensionality would result in an increase of complexity and therefore a decrease of interpretability of the model. Neurosymbolic approaches are a response to this problem as they combine the interpretability of symbolic approaches with the state-of-the-art performance of machine learning models. In parallel, self-explainable models are being developed with the same goal of providing high performance and high interpretability. Such interpretable models still require the generation of an explanation that is adapted to the user. To the best of our knowledge, there is no explainable AI system capable of predicting and generating explanations tailored to the user. A generic architecture of an explainable intelligent system (XIS) was proposed by DARPA to give a direction to the research in XAI [15] (see Chapter 5 for further details about this architecture). Self-explainable and neurosymbolic models are the closest to this architecture but do not include one of the most important part that is the system that generates and presents the explanations to the user.

These observations motivate the design of a complete XIS that entirely follows DARPA's pro-

posed architecture. This design should follow a set of good practices to avoid the pitfalls detected in the literature that are the ambiguous terminology and the lack of adequate evaluation. Moreover, we argue that the goal of explaining to control or improve are poorly represented in the current XAI landscape. Hence, the XIS should be able to confirm or deny its predictions and explain why. We hypothesize that a neurosymbolic approach could be used for that purpose. Particularly, we intend to explore the combination of machine learning models with Semantic Web Technologies (e.g. ontologies or knowledge graphs) as the latter have been identified as ideal candidates to solve current XAI challenges [31]. It was mentioned that interpretable models should be associated with explanation methods that leverage their interpretable nature. That is why an explanation method specific to our proposed neurosymbolic model will also be developed in this thesis.

2.3.2 Contributions and outline

This thesis presents three main contributions.

1. A terminology of XAI, made to remove any ambiguity in the definitions while remaining compatible with the majority of the literature. This terminology contains the definition and composition of an explanation along with the definition of reoccurring terms found throughout the XAI literature that quality AI systems.

2. The design of an XIS based on DARPA's architecture [15]. This proposed XIS is decomposed into two parts: a novel neurosymbolic model for classification and an explanation interface that leverages the model to generate adequate explanations. The neurosymbolic model uses an ontology to create and train machine learning models and check the consistency of the predictions made by these models. Therefore, it is capable of warning the user when a prediction is inconsistent with expert knowledge as well as explain why. An explanation interface is then developed to present the results of the XIS to the user.

3. A method to generate counterfactual explanations for ontologies. This mode of explanation is especially adapted to explain the prediction and consistency detection of the aforementioned XIS. Moreover, it is designed to be applicable to most existing ontologies as a tool to debug and repair them. This method is inspired from existing methods to generate such explanations for machine learning. It uses the same principles but several problems specific to ontologies arise that are solved in this contribution.

The chapters that present each contribution contain the necessary state of the art to identify the existing solutions and their limitations. The thesis is organized as follows:

Chapter 3 presents the necessary background for this thesis. A first part introduces an overview of the XAI landscape. The three main problems and the corresponding solutions proposed by the community are discussed. The first problem is about the definition of an explanation, especially the components that compose an explanation. The second problem is the design of XAI methods and the creation of taxonomies to classify these methods. The last problem is the evaluation of explanations and the identification of criteria to evaluate the quality of an explanation. The solutions to these three problems are discussed and their limitations are identified. The second section provides the technical background about ontologies that are used in this thesis. Firstly, the notion of ontology is introduced along with a more general discussion about symbolic AI. Afterwards, Description Logics are described and finally, the Web Ontology Language (OWL) is introduced.

Chapter 4 describes our first contribution which is a terminology for XAI that is used to define the terms in the remainder of the thesis. It introduces reoccurring technical terms in the literature and proposes unambiguous definitions that are also compatible with the majority of the definitions encountered in the literature. An ontology that represents the relation between each term is created to facilitate the comprehension of the terminology.

Chapter 5 presents the second contribution: the design of an XIS capable of detecting inconsistencies in its predictions and explaining why they were detected. Specifically, the creation of a neurosymbolic model that combines an ontology and machine learning models is depicted. Then the design of an explanation interface that presents the results of the model along with an explanation of why the prediction is consistent or not is discussed.

Chapter 6 introduces the third contribution which is the creation of a method to generate counterfactual explanations for ontologies. A first part is dedicated to understanding the nature of counterfactual explanations and the existing approaches proposed for machine learning. The second part presents our approach to design these explanations specifically for ontologies.

Chapter 7 presents the experiments performed to validate and evaluate the proposed XIS and the generation of counterfactual explanations for ontologies. Firstly, a test case is depicted that is the classification of a musical instrument in an image. The dataset and ontology corresponding to this case are further discussed. Then, experiments to assess the validity of the neurosymbolic model and its ability to detect inconsistencies are performed. Finally, a small scale user study is conducted on the musical instruments classification task with a group of expert users to evaluate the validity of our approach. The predictions are made by the XIS and the results along with counterfactual explanations are presented to the users who are asked to evaluate the relevance of the provided explanations.

Chapter 8 concludes this thesis with an analysis of the presented contributions and a discussion on the perspectives for future work.

2.4 Publications

The following works were published in peer-reviewed conferences and journals during this thesis:

- **Bellucci M.**, Delestre N., Malandain N. and Zanni-Merk C., (2021), *"Towards a terminology for a fully contextualized XAI"*. In Proceedings of the 25th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, KES 2021, 8-10 September 2021, Szczecin, Poland. DOI: `10.1016/j.procs.2021.08.025`

- **Bellucci M.**, Delestre N., Malandain N. and Zanni-Merk C., (2022), *"Une terminologie pour une IA explicable contextualisée"*. In Conférence francophone sur l'Extraction et la Gestion des Connaissances (EGC 2022) 2022 as part of EXPLAIN'AI Workshop, 24-28 January 2022, Blois, France.

- **Bellucci M.**, Delestre N., Malandain N. and Zanni-Merk C., (2022), *"Ontologies to build a predictive architecture to classify and explain"*. In the European Semantic Web Conference 2022 (ESWC 2022) as part of the Deep Learning meets Ontologies and Natural Language Processing (DeepOntoNLP) Workshop, May 29 - June 2 2022, Hersonissos, Greece.

- **Bellucci M.**, Delestre N., Malandain N. and Zanni-Merk C., (2022), *"Combining an explainable model based on ontologies with an explanation interface to classify images"*. In Proceedings of the 26th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, KES 2022, 7-9 September 2022, Verona, Italy. DOI: `10.1016/j.procs.2022.09.298`

Chapter 3

Background on XAI and Symbolic AI

Contents

3.1 Background on XAI 31
3.1.1 Defining an explanation 32
3.1.2 Designing explainability methods 35
3.1.3 Evaluating XAI 38
3.1.4 Limitations 47
3.2 Background on ontologies 49
3.2.1 Ontologies and their applications 50
3.2.2 Description logics 52
3.2.3 The Web Ontology Language (OWL) 53
3.2.4 Discussion 56

Machine learning models suffer from a lack of explainability that hinders their application in sensitive domains. The problem of explaining the decisions made by an AI system also arose for symbolic AI methods. However, the nature of symbolic AI facilitated the generation of explanations due to the use of deductive reasoning and human-understandable symbols. Hence, the domain of XAI has not been heavily researched until very recently. To find solutions to the lack of explainability, some scholars explore the combination of symbolic AI methods with machine learning models. Most notably, ontologies seem to be ideal as they can both represent and reason about data in a way that is both machine-readable and human-understandable. This thesis follows this research direction and seeks to address open problems in XAI with symbolic approaches.

In this chapter, we review the literature of XAI in order to identify the research directions, the limitations of the existing methods and open problems. Afterwards, we introduce the technical foundations of ontologies to equip readers with the necessary knowledge to understand the work conducted in this manuscript.

3.1 Background on XAI

Explaining is a key human behavior that is heavily studied from different angles by various scientific fields e.g. philosophy, psychology or sociology. The mechanisms of an explanation made by humans are not fully understood. Yet, the XAI domain aims at automatically formulating explanations about AI systems. This requires a way to generate an explanation that meets the standards set by humans, as well as being able to understand and transcribe the decision process of the AI system. Hence, the problem that XAI is set on tackling necessitates collaboration among various scientific communities from AI to humanities, resulting in works that are written from different points of views and different goals.

This section is a review of the XAI domain that aims at giving a good understanding of the current state of the XAI literature and its unaddressed problems. It is divided into three parts that

correspond to the three main problematics of the field. The first part is about the identification of the definition and the components of an explanation which would streamline the generation of explanations. The second part introduces the different categories of explainability methods that are observed in the literature. Finally, the third part discusses the evaluation of the explanations and explainability methods.

3.1.1 Defining an explanation

Although the first attempts at a definition of what an explanation is date back to Ancient Greece, it remains an open discussion in contemporary days [66]. The definition of explanation changes over time but also according to the discipline that uses it. Indeed, each science field explains for different purposes as Tiddi et al. [65] illustrated in an analysis of explanations. The field of XAI has seen numerous attempts at defining an explanation ([28, 32, 45, 67]) which demonstrates the difficulty to propose a consensual definition that captures their diversity and complexity. Knowing what constitutes an explanation and what makes an explanation effective is of utmost importance. It would streamline the generation and evaluation of explanations and drastically speed up progress in XAI.

The definition of explanation according to the Cambridge dictionary is *the details or reasons that someone gives to make something clear or easy to understand.* An explanation is therefore a relevant set of information along with some kind of reasoning, that will enable the explainee to understand. Cabitza et al. [32] propose three criteria that make an explanation: the explanandum (the thing to be explained), the explanans (an argument, fact or sign) and an explanatory relationship that holds between the explanans and the explanandum. This explanatory relationship connects the explanans and the explanandum. In the case of deductive reasoning, the explanatory relationship expresses a logical consequence where the explanans is a premise and the explanandum is a consequence. Tiddi et al. [65] share a similar but more detailed view about the constituents of an explanation. An explanation links two events, the explanans and explanandum that are set in a particular context. This link is based on a set of assumptions that they call theory and is similar to the explanatory relationship.

However, as Cabitza et al. [32] point out, explainees are not explicitly represented in their explanation structure. The prior knowledge, experience and beliefs of the explainee should have an impact on the explanation [34, 35]. Miller [33] insists on the social nature of explanations, they should be presented relative to the explainer's beliefs about the explainee's beliefs. Other scholars [37, 46, 68] corroborate the necessity to include the explainee in the design of an explanation. Similarly to the previous definitions, they argue that an explanation should use causal reasoning i.e. identify the causes that led to the effect to be explained using some kind of reasoning that is not necessarily deductive. In addition, they describe what methods of reasoning are used and how they are influenced by the explainee. Individuals require different causal reasoning and expect different causes based on their domain and level of expertise. For simplicity, scholars propose three categories of explainee:

AI experts They are interested in technical explanations that allows them to debug and improve the explained AI system. They have a good understanding of how the AI system functions and are able to comprehend sophisticated and technical explanations.

Domain experts They are experts in the application field of the explained AI system. They are interested in understanding the causes of a particular decision and assessing the accuracy of the system. They may also need a detailed explanation to then inform others (e.g. customers or patients).

Laypersons They do not have a particular expertise in AI or in the application field of the AI system. Like the domain experts, they are interested in understanding the causes of a particular decision but require simple and non technical explanations.

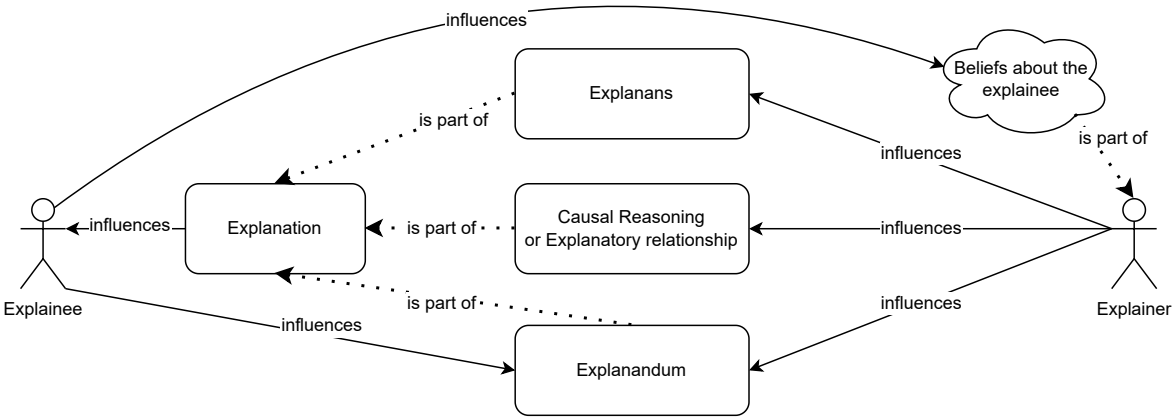


Figure 3.1: Diagram of an explanation. Arrows represent a direct influence from the source on an element (e.g. the explainer chooses the explanans). Dotted arrows represent a component (e.g. explanans is a component of an explanation).

To summarize what constitutes an explanation, we propose in Figure 3.1 a diagram inspired from Cabitza et al. explanation diagram [32] extended with the notions of explainee and explainer. We use explanatory relationship and causal reasoning as synonyms for this diagram. An explanation is an interaction between two agents: the explainee and the explainer. Usually, this interaction is started by the explainee with a question about the explanandum, therefore the explainee has a direct influence on the explanandum. The explainer formulates an explanation to answer the explainee. As previously discussed, the explanation is composed of the explanandum, the explanans and an explanatory relationship or causal reasoning to connect them. Ultimately, it is the explainer that formulates the entire explanation and thus chooses the causal reasoning, the explanandum and explanans. Miller [33] argued that the explanation depends on the explainer's beliefs about the explainee, i.e. what the explainer assumes about the explainee. For instance, the explainer will not formulate the same explanation if they believe that the explainee is an expert of the domain instead of a layperson. Evidently, the explainee has a direct influence on such beliefs and is influenced by the explanation.

Multiple choices are possible for each component of the explanation. Concerning the explanandum, questions asked by the explainee can be about two categories: the outcome or group of outcomes from the AI system or the functioning of the entire AI system. The former category is called local, the latter is called global. Regarding the explanans, many possibilities are being explored such as feature relevance (or importance), examples, counterfactual cases or rules. These categories are explored in Section 3.1.2. Finally, the causal reasoning or explanatory relationship is little studied in the state of the art. As Durán [36] points out: "If we were to ask partisans of XAI what they have to say about the explanatory relation, we would probably hear very little". Indeed, XAI methods tend to only provide explanans and let the explainee make the relationship between the explanans and explanandum. There are several forms of causal reasoning that humans use. Deduction and induction are mostly seen in a scientific context. Deduction is applied for demonstrations, it utilizes logic to draw a conclusion based on a set of premises. A deductive inference is always true when the set of premises is true. Producing explanations for AI systems based on deductive reasoning is complex, the explanans and explanandum should be represented in the language of first-order logic [36]. Moreover, because of the complexity of machine learning models and their inductive nature, it is not always possible to propose a deductive reasoning to link explanans and explanandum. Nevertheless, rule-based or knowledge-based systems apply deductive reasoning to make decisions and predictions thus rendering deductive explanations possible. Induction is another form of reasoning that draws a conclusion based on a limited set of cases. In contrast to deductive reasoning, the conclusion derived from the cases is not guaranteed to be correct. Hoffman et al. [37] describe induction as equivalent to "generalization". However, to make such a generalization, an undefined number of observations must be acquired, which may

Type of reasoning	Definition	Example
Deduction	The process of making an inference based on accepted laws and logic.	It rains outside (cause) therefore they took their umbrella (effect)
Induction	The process of inferring the cause of an effect, based on a set of observations.	They took their umbrella (effect) every time it rained outside (observation), therefore the rain causes them to take their umbrella (identified cause)
Abduction	The process of identifying the most probable cause to an observed effect.	They took their umbrella today (effect), it is probably be raining outside (most plausible cause).
Counterfactual	The process of identifying necessary causes by hypothesizing what would happen to the effect if some cause were different.	If it hadn't been raining outside (counterfactual hypothesis), they wouldn't have taken their umbrella (hypothesized effect). Therefore rain is a cause of them taking their umbrella.

Table 3.1: A summary of methods for causal reasoning.

not be ideal to explain a single prediction. Abduction is a reasoning method that finds the most probable cause to some effect. Miller [33] describes the process of abduction as the following: (1) observe some event; (2) generate one or more hypothesis about the event; (3) judge the plausibility of the hypotheses; and (4) select the "best" hypothesis as the explanation. Some scholars argue that abductive reasoning is closely related to explanation and therefore particularly relevant to generate explanations [33, 37, 38]. Despite the relation between abduction and explanation, few works in XAI explicitly use abductive reasoning to generate explanations. Finally, counterfactual reasoning is a form of reasoning that is gaining traction in the XAI community. Stepin et al. [61] discuss counterfactual[1] reasoning as the basis of abductive inference. Studies in social sciences show that determining a cause-effect relation usually calls for counterfactual (or contrastive) reasoning [33, 37, 61]. Counterfactual reasoning is an exploration of the influence that some causes have on the effect to be explained. This type of reasoning allows to identify a set of causes that are necessary to produce the effect. Wachter et al. [39] triggered a new trend in the XAI community by proposing a method to generate counterfactual explanations, arguing that this type of explanation is technically feasible and particularly relevant. Table 3.1 sums up the different types of reasoning.

We emphasize the fact that the definition of explanation we discussed is not consensual. For instance, Durán [36] argues that the explanation should not always depend on the explainee, especially for scientific explanations. Others consider that the output of an XAI method (i.e. the explanans) is an explanation and do not provide an explanatory relationship between the output of the method and the explanandum. Still, scholars agree that the goal of an explanation is to help the explainee identify the reasons of a decision or an event. As a result, it is difficult to evaluate the quality of an explanation. Intuitively, an explanation is good when the explainee is satisfied by the explanation and considers that they have understood the causes of the explained event. Yet, only considering the level of satisfaction of the explainee may lead to dangerous implications, as the explainer could formulate a misleading but persuasive explanation. In the context of Responsible AI, this possibility should be avoided and different criteria should be added to ensure that the explanation reflects as much as possible the actual causes of a decision or event. These criteria are further discussed in Section 3.1.3.

[1] We use counterfactual as synonym of contrastive although these two notions are slightly different (see [61]).

3.1.2 Designing explainability methods

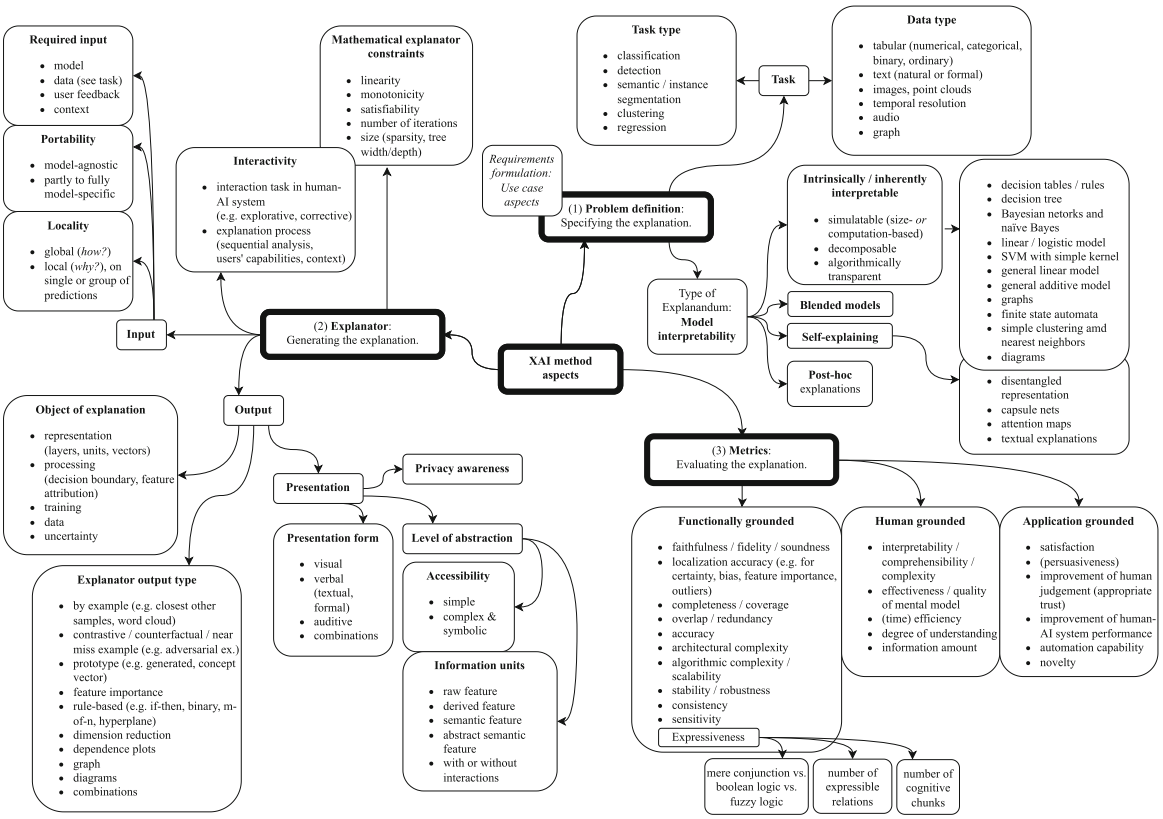
In Section 3.1.1, we discussed the components of an explanation and the many options available for each component that partially reflects the diversity of explanation methods. Other notions come into play when selecting and creating explanation methods that we will study in this section. In an attempt to identify these notions and consequently categorize XAI methods, many taxonomies have been created over the years [44]. These taxonomies are not consensual, mostly due to the fact that the classification of XAI methods depends on their purpose and audience. Schwalbe and Finzel [69] undertook the creation of a comprehensive taxonomy by analyzing over 70 surveys on XAI and identifying overlapping concepts. However, the resulting taxonomy is complex, with over 20 different categories each containing several classes, and is intended for XAI experts. A different approach to building taxonomies is by choosing the categories for a particular focus. In a paper aimed towards helping newcomers to the field of XAI, Speith [44] presents four approaches to building XAI taxonomies and proposes a combination of these taxonomies. The four approaches are: *functioning-based approach* that categorizes the different ways XAI methods extract information from AI systems; *result-based approach* that classifies the types of output from XAI methods; *conceptual approach* that distinguishes different dimensions that make up an XAI method (e.g. the scope of the explanation or the type of problem such as classification or regression); and (4) *mixed approach* that is a hybrid of the three other approaches. Figure 3.2 illustrates the difference in complexity between the comprehensive taxonomy of Schwalbe and Finzel (Figure 3.2a) and the less detailed taxonomy destined for newcomers (Figure 3.2b). It depicts the wide diversity and complexity of XAI methods which prevents newcomers from acquiring an adequate picture of the XAI landscape [44]. In the same spirit of rendering XAI more accessible to newcomers, Arya et al. [34] provided a decision tree to help AI practitioners and non-experts choose the most adequate XAI method for their needs.

Despite the lack of consensus and difficulties to establish a unique taxonomy, some distinctions between XAI methods are commonly discussed. Notably, three categories of methods are systematically referenced. *Global/local* explanations that refer to what is being explained i.e. either explain the functioning of the AI system for global explanations or explain a particular prediction for local explanations [29, 34, 44, 45]. The methods to generate the explanations are usually divided into two categories, *post hoc* or *ante hoc*[2] explanations. The former contains methods that explain a trained model [44] using an auxiliary method to create the explanation [34]. The latter exploits the functioning of the model to derive an explanation thanks to its interpretable or transparent nature [44, 46]. Finally, the distinction between *model-specific* and *model-agnostic* methods is usually made in taxonomies.

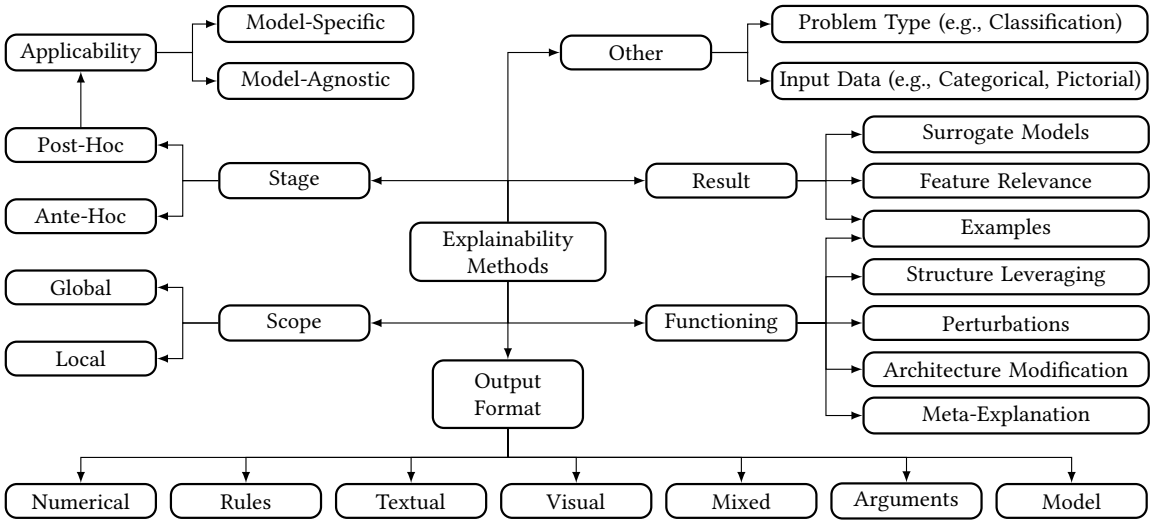
As reflected by result-based and functioning-based taxonomies, XAI methods provide different forms of results by using different mechanisms. Three main categories of results are identified by Speith [44]: *feature importance*, *surrogate models* and *examples*. *Feature importance* (or feature-relevance) methods calculate an importance score for each feature to show the users the impact of each feature on a prediction [28]. *Surrogate models* are usually directly interpretable models that approximate the behavior of the model to explain, enabling the use of ante hoc methods or visualization of this surrogate model [34, 44]. Finally, *example-based* methods propose representative examples that explain the behavior of the model [29, 44].

The first popular class of XAI methods create a local surrogate linear model to extract feature importance for a particular prediction. In this class, we find the well-known methods LIME [49], DeepLIFT [51], LRP [70] or SHAP [50]. They all share the following framework that is *additive feature attribution*. This framework consists in finding a simplified input x' which is a binary vector where each feature is interpretable. This simplified input is associated to a mapping function h_x so that $h_x(x') = x$ where x is the original input. Then the weights of the surrogate linear model are selected under the constraint that the surrogate model must accurately replicate the original model in the neighborhood of the prediction to explain i.e. for $z' \approx x'$, $g(z') \approx f(h_x(x'))$ where

[2] Ante hoc methods are also called *transparent models* but the name is subject to debates in the literature [44]



(a) Schwalbe and Finzel's taxonomy [69]



(b) Speith's taxonomy [44]

Figure 3.2: A comparison of two taxonomies of XAI methods

f is the original model and g the surrogate. This class of local XAI methods can be turned into a global method as illustrated by SP-LIME [49] where a set of representative examples are selected and the local method is applied on this set of examples in order to explain the behavior of the entire system. However, Rudin [71] warns that post hoc methods, provide explanations that are not faithful to the original model.

Faithfulness, also named correctness, truthfulness or fidelity is an important criterion for explanations as it ensures that the explanation accurately reflects the model's decision process [72, 73, 74]. Indeed, a persuasive natural language explanation can be easily generated by AI systems such as ChatGPT, without being grounded with the actual AI system to explain. What makes an explanation relevant is its ability to correctly identify the causes of a prediction, based on the functioning of the system. Using simplified features and a simplified model severely impacts the faithfulness and thus the quality of the resulting explanation. To overcome the problem of faithfulness, new interpretable or self-explaining models are proposed. For instance, Alvarez-Melis and Jaakkola [75] proposed Self-Explaining Neural Networks (SENN), a method also inspired by interpretable linear models but that avoids the pitfalls of the previously discussed methods. In this paper, they identify the desired properties of interpretable models and generalize them in order to create a neural network with these properties while retaining the advantages of neural networks. The main advantage of interpretable or self-explaining models is that explanations are always truthful, since no post hoc method is applied to explain. Furthermore, many methods simplify the input features into human understandable features to increase the system's interpretability. SENNs [75] use a mapping function from raw features (e.g. pixels in an image) to *interpretable basis concepts* (e.g. basic shapes). This function can be either an aggregate of the raw features or a predefined feature extractor designed with expert knowledge or a learned representation. The TCAV method [76] goes further by enabling the user to quantify the influence of a concept on a prediction. For instance, the user can quantify the influence of "stripes" on the prediction of the class "zebra" and make sure this influence is coherent with the knowledge that zebras have stripes. There is a clear need for the use of human-understandable concepts as input features or in the explanations to ensure the quality of the explanation. Likewise, we have seen that faithfulness is ensured when interpretable models are used, since they are sufficient to generate explanations. Interpretable models that use human-understandable concepts remind of expert systems and more generally of symbolic AI.

We discussed in Chapter 2 that the use of symbolic AI methods in combination with machine learning models could lead to AI systems that are explainable and share the same level of performance as the current models. Calegari et al. [77] studied the integration of symbolic approaches with sub-symbolic (i.e. numerical, statistical and distributed representation of machine learning models) for XAI. They distinguish two categories of methods that combine symbolic and sub-symbolic: integration and composition. Integration methods are in fact neurosymbolic, they use both logic and statistical or numerical approaches to learn and predict. For instance, Logic Tensor Networks (LTN) [78] integrate first-order logical constraints to a neural network so that reasoning about the constraints improves learning and inversely, learning from new data can revise the constraints. The logical constraints are integrated in a loss function used to train a tensor network with the task of approximating the truth value of the constraints given as input. Another example of integration techniques is the neural theorem prover (NTP) [79] which builds a neural network recursively, based on a backward-chaining reasoning algorithm. NTP replicates the backward-chaining reasoning algorithm by using modules that represent a logical operation of this algorithm (unification, AND, OR). These modules take symbols as input (e.g. atoms and rules) and return a proof success score to evaluate the proof. They are chained up to form the final network capable of automatically completing a knowledge base. On the other hand, composition methods keep symbolic and sub-symbolic approaches in identifiable separate blocks that cooperate to produce a prediction. Knowledge extraction methods are similar to surrogate models, they attempt to extract the knowledge learned by a machine learning model and represent it in a symbolic form (mostly rules or decision trees). TREPAN Reloaded [80] uses the existing TREPAN method [81],

a method that extracts a decision tree as a surrogate model for a trained neural network, and exploits knowledge in the form of an ontology to choose the splitting nodes. Therefore, the surrogate decision tree uses human-understandable concepts which improves the overall interpretability of the method. Conversely, knowledge injection methods aim at embedding parts of a knowledge base into continuous vector spaces so that neural networks can leverage background knowledge to perform machine learning tasks. The combination of ontologies or knowledge graphs with machine learning models to gain explainability is also being explored by researchers. It is argued that machine learning and ontologies can improve each other but can also be used for explainability [46]. Moreover, ontologies provide a huge potential in making complex data structures understandable and are identified as a key to achieve truly explainable AI [46, 82]. Indeed, ontologies and knowledge graphs are large-scale structured representations of data and knowledge that allow for logical reasoning about this knowledge. Lécué [31] reviews XAI methods in various fields and how their current limitations can be addressed by knowledge graphs. The use of ontologies for XAI is further motivated by their wide adoption for linking data on the Web, as illustrated by the knowledge graphs of Google, IBM or Microsoft [83] along with collaborative projects like Wikidata [84] or ConceptNet [85]. Several surveys on the use of ontologies for XAI have been carried out, showing overall that there is a gain in explainability without reducing the performance of the models [82, 86]. However, Seeliger et al. [82] exposed the lack of model-agnostic methods and the concentration of knowledge-based XAI methods on specific machine learning techniques, especially neural networks. Tiddi and Schlobach [86] discussed the technical challenges of using large-scale knowledge graphs, notably the computational cost of exploiting these graphs and the need to efficiently extract relevant knowledge as all reviewed methods require a manual selection of information from the graph.

To conclude this review of XAI methods, we have seen that XAI is being actively developed, with hundreds of papers being published every year since 2010 [28, 87]. Several categories of XAI methods have been identified, particularly the distinction between post hoc and ante hoc methods. Post hoc methods suffer from several issues originating from their nature, which is to use an auxiliary method to explain instead of using the functioning of the original model. The most important issue is the lack of faithfulness of the explanation, i.e. whether the explanation reflects the actual decision process. Ante hoc methods are naturally interpretable which guarantees faithfulness. However classical interpretable methods such as linear models are considered less accurate, which led to the creation of new classes of models inspired from interpretable models. The development of neurosymbolic approaches is therefore a logical step to develop new interpretable models that can achieve state-of-the-art performance. Although Calegari et al. [77] mention the possible use of neurosymbolic approaches for XAI, very few neurosymbolic methods are explicitly developed to be interpretable or explainable. Therefore these methods do not focus on explainability and adapting them to generate explanations may require additional work. It is clear that neurosymbolic methods are focusing on improving the performance of AI rather than its explainability. The integration of Semantic Web Technologies to current AI algorithms for XAI is being actively discussed but research in this direction is not well explored with few methods being proposed every year. Ontologies and knowledge graphs have clearly been identified as good candidates for XAI thanks to their ability to represent and reason about large-scale structured data. Unfortunately, research in XAI is stalled by the lack of standard benchmarks and user-studies to evaluate the quality and relevance of the proposed methods.

3.1.3 Evaluating XAI

Evaluating a new method or model is a standard and mandatory practice to validate and assess its quality and relevance. A set of benchmarks is usually created to standardize the evaluation of some class of methods which enables the comparison between them. Unfortunately, in XAI, evaluation is often overlooked and no standard benchmarks have been accepted [88]. Indeed, Adadi et al. [29] reported that only 5% of the papers they reviewed focused on evaluating and quantifying the relevance of XAI methods. Likewise, Nunes and Jannach observed that 21.5% of the reviewed

XAI methods contained a relevant form of evaluation [89]. This phenomenon can be explained by the lack of consensus in the definition of an explanation and specifically the identification of criteria that make a good explanation. The identification of such criteria and how to accurately quantify them is heavily discussed in papers focused on solving the evaluation problem of XAI. Explanations are a social process [33], which implies that the identification of subjective criteria is unavoidable, therefore a distinction is made between objective and subjective criteria. Objective criteria relate to criteria that do not depend on the user, such as the fidelity of an explanation. On the other hand, subjective criteria describe elements that should be adapted to the user's needs and preferences and require a human to be measured. Consequently, the evaluation of these criteria need to include user-studies that may prove costly and difficult to get right. User-studies are therefore mostly avoided, Nauta et al. [42] report that the amount of papers including a user-study is constant over the years and represent around 20% of the literature of XAI methods. In response to this problem, proxy metrics are designed to approximate the evaluation of subjective criteria without requiring a user-study. Overall, the evaluation of XAI faces two main issues: 1) identifying criteria of a good explanation and 2) design methods to efficiently quantify these criteria including the design of proxy metrics that avoid the need for user-studies.

Identifying the criteria of an explanation

We studied several propositions of criteria from the literature. Considering the number of papers that discuss and/or propose criteria of an explanation ([13, 40, 41, 42, 43, 74, 90, 91, 92, 93]), the following review is not exhaustive but is representative of the current state of the art. Three main criteria are identified by Yang et al. [40]: *generalizability*, *fidelity* and *persuasibility*. *Generalizability* is an indicator of the generalization performance of a set of explanations. A generalizable explanation allows the user to anticipate the outcome of the system in different situations rather than just the explained one. *Fidelity* relates to the ability of an explanation to precisely capture the decision making process and show the correct evidences. It is also known as faithfulness or correctness. The third criterion described by Yang et al. is *persuasibility* which corresponds to how human comprehend and respond to an explanation. It measures the satisfaction or comprehensibility of an explanation for a particular group of users. They argued that *persuasibility* is the only subjective criterion that varies depending on the users and tasks. Zhou et al. [41] identified two main criteria of explainability that are subdivided into several properties. According to them, explainability is composed of *interpretability* and *fidelity*. *Interpretability* relates to the subjective part of an explanation i.e. how understandable an explanation is. *Clarity, parsimony* and *broadness* are the three properties of *interpretability*. *Clarity* measures whether an explanation is unambiguous, *parsimony* corresponds to the simplicity and compactness of the explanation and *broadness* describes how generally applicable an explanation is. The other criterion of explainability is *fidelity* which shares the definition proposed by [40]. *Fidelity* is divided into two properties: *completeness* assesses that the explanation describes the entire dynamic of the model; *soundness* regards how correct and truthful the explanation is. We note similarities between the criteria identified by Yang et al. and those proposed by Zhou et al.. *Clarity* and *parsimony* broadly correspond to the notion of *persuasibility*, *broadness* and *completeness* match with the *generalizability* criterion and *soundness* with *fidelity*. Förster et al. [43] proposed a set of criteria of an explanation rooted in studies from social science. They identified that *shortness, coherence, generality* and *relevance* are key characteristics of explanations. *Shortness* refers to the number of causes invoked and is similar to the concepts of *parsimony*. *Coherence* describes the ability of an explanation to be consistent and relate to prior beliefs of the explainee. *Generality* depicts the ability of explanations to explain more events thus matching with the notions of *generalizability* and *broadness*. Finally, Förster et al. discussed the *relevance* of an explanation which refers to the choice of relevant causes. They consider a cause relevant if it refers to situations that are not too far in the past, surprising or abnormal. We argue that *coherence* and *relevance* are part of the *persuasibility* notion, as they participate in the satisfaction and comprehensibility of an explanation which corresponds to the definition of *persuasibility* given by Yang et al. [40].

Most works focused on identifying criteria of an explanation describe these criteria as absolute ideals to reach e.g. an explanation needs to be short to be understood or needs to be generalizable to be satisfying. Zhou et al. [41] said that the proposed criteria need to be satisfied to achieve explainability. This view is implicitly shared in many works as they are usually based on intuition of what an ideal explanation should be. Miller et al. [11] warned about this phenomenon that they described as "inmates running the asylum". They noted that a large proportion of XAI methods were evaluated based on the authors' ideal of explanation and not data driven characteristics. Förster et al. [43] discussed the fact that the ideal value of some criteria depend on the user's preferences and the task. They observed that short explanations are generally preferred by humans, however longer explanations are sometimes required in situations such as a scientific explanation. It appears that the ideal value of some criteria depend on the context of the explanation such as the length (or shortness) or the generalizability while other criteria have constant ideal values independent of the user or context like fidelity or clarity. Yet, criteria of an explanation may be contradictory and imply trade-offs. Particularly, Nauta et al. [42] identified 12 dimensions or criteria of an explanation that contain implicit trade-offs. They called these dimensions *Co-12* since they named each dimension with a word starting with "Co".

Correctness is similar to *fidelity* or *soundness*, it measures the descriptive accuracy of an explanation.

Completeness describes how much of the model's behavior is explained by a single explanation. Zhou et al. [41] share the naming and definition of this criterion while Yang et al. [40] include *completeness* in the broader notion of *generalizability*. They discussed reasoning-completeness and output-completeness. Reasoning-completeness indicates how much of the internal dynamic of the model is described, ranging from using a global surrogate model that does not explain the internal reasoning of the actual model to revealing all the mathematical operations of the system. Output-completeness quantifies how well the explanation agrees with the predictions of the original model. Using a surrogate model systematically reduces the *correctness* of the explanation. Likewise, providing a complete explanation requires a lot of information that directly impacts the *compactness* of the explanation. Hence Nauta et al. acknowledged that *completeness* should be balanced with *compactness* and *correctness* to avoid overwhelming the explainee with too much information.

Consistency checks that identical inputs have identical explanations. Models that give the same outputs for all inputs should be explained with the same explanations. We argue that *consistency* helps the user trust an explanation because a person expects that a same event always has the same causes. Therefore it plays a role in the *persuasibility* of an explanation.

Continuity is equivalent to the notion of stability in mathematics, it ensures that small variations in the input do not lead to large changes in the explanation. It can be seen as an extension of consistency, similar inputs should have similar explanations. Several authors also use the term *robustness* as it is similar to the problematic of robustness in machine learning and is commonly used to evaluate *fidelity* [94, 95] while Nauta et al. argued that *continuity* participates to *generalizability*.

Contrastivity reflects the idea that an explanation should answer "why not?" or "what if?" questions. It carries the idea that an explanation should contain information specific to a target or event. When explaining a system that detects cats and dogs in an image, the explanation of why a dog has been detected should include specific features of a dog. An explanation saying that paws and fur were detected is not contrastive since this explanation is also valid for a cat.

Covariate complexity refers to the comprehensibility of the relation between the covariates (i.e. features) and the target to explain. This includes the interpretability of the features presented. Human-understandable concepts should always be preferred even if it does not correspond to the actual inputs of the model. The interactions between the presented features

and the target should also be described with simple functions respecting some properties such as monotonicity, in order to be quickly and easily understood by humans.

Compactness is heavily documented in the literature. It addresses the size of the explanation and contains the criteria of sparsity, shortness and redundancy. Although Nauta et al. mentionned that explanations should be sparse and short, we have seen that Förster et al. [43] qualified this requirement with their study showing that the ideal *shortness* depends on the goal of the explanation. Nauta et al. argued that there is a trade-off between *completeness* and *compactness*, implying that some users may prefer complete explanations rather than compact ones.

Composition concerns how the explanation is presented e.g. the presentation format, organization and structure of the explanation. Choices about the representation of the explanation, the terminology or the usage of higher-level information are included in order to increase interpretability and *clarity*[3].

Confidence regards the inclusion of a probability or measure of certainty in the explanation. This measure can determine either the confidence of the model's prediction or the likelihood of the explanation, although providing the likelihood of the explanation in the form of a probability reportedly divides the community since people have difficulties to correctly estimate probabilities.

Context addresses whether the user and their needs are taken into account to generate comprehensible explanations. The explanation should be adapted to the user's needs and their expertise as discussed earlier. This notion is close to the idea of *relevance* discussed by Förster et al. [43], explanations should refer to meaningful situations that depend on the context of the explanation.

Coherence depicts how much the explanation is consistent with the user's background knowledge and beliefs i.e. the plausibility of an explanation. It is noted that this property is limited to external coherence (i.e. explanation is coherent with the user's beliefs), in contrast to internal coherence (i.e. the parts in the explanation fit together). This definition of *coherence* is highly similar to the one from Förster et al., though the latter includes both external and internal *coherence* in their definition.

Controllability describes the ability of an explanation to be controlled, corrected or interacted with a user.

Choices concerning these criteria must be made to generate an ideal explanation. Choices concerning criteria related to the complexity of an explanation are especially important to design the best explanation. Indeed, completeness, correctness, covariate complexity and compactness all impact the complexity of an explanation. A compact explanation will not carry enough information to be complete or capture the actual relations between features and target leading to an impact on covariate complexity and correctness. Moreover, we argue that all these criteria are interconnected, improving one criterion may deteriorate another, thus making it nearly impossible to create a perfect explanation. Hence the need for controllability or interactivity that is promoted by many scholars [11, 15, 34]; being able to control, correct and interact with the explanation will help the user design their own custom explanation. This also explains the wide variety of forms of explanations, researchers are looking for the perfect explanation but it doesn't exist.

Table 3.2 shows the different criteria identified in [40, 41, 42, 43] that we discussed and illustrates the relations between these criteria, based on the definitions provided by the authors. For comparison, similar tables are also proposed in [90, 96] with different sets of articles. Some criteria defined in a paper may encompass several others from another paper. Particularly, the criteria proposed by Yang et al. [40] are very broad and can be divided into many sub-criteria as illustrated

[3] Nauta et al. [42] directly cited the notion of *clarity* proposed by Zhou et al. [41] when discussing *composition*.

Yang et al. [40]	Förster et al. [43]	Zhou et al. [41]	Nauta et al. [42]
Generalizability	*Generality*	*Broadness*	Continuity
	-	*Completeness*	*Completeness*
		-	Consistency
			Covariate complexity
Persuasibility	*Shortness*	*Parsimony*	*Compactness*
	-	Clarity	Composition
			Covariate complexity
	Relevance		Contrastivity
		-	Consistency
	Coherence		Coherence
			Context
Fidelity	-	*Soundness*	*Correctness*
		Completeness	*Completeness*
		-	Confidence

Table 3.2: criteria of an explanation as seen by different authors. Criteria on the same line and in *italics* share near-identical definitions.

by this table. We note that some aligned criteria do not have the same granularity e.g. *continuity* is a sub-criterion of *generality* or *broadness* meaning that it is a tool that can help providing general explanations. Likewise, the definition of *coherence* from Förster et al. [43] is broader than the one from Nauta et al. [42], the latter only ensures that the causes are coherent with prior knowledge and beliefs of the user while the former adds the need for the consistency of the explanation. Confusingly, consistency in the definition of *coherence* does not seem to have the same meaning as the notion of *consistency* proposed in [42]. Förster et al. use the definition of consistency from logics: the explanations should not contain internal contradictions (as defined in [97]) whereas Nauta et al. define consistency as deterministic: identical inputs should have identical explanations. Hence, *relevance* is paired with *consistency* since Förster et al. say that relevant explanations should not refer to surprising situations. If different causes are given for the same event at the same time, we argue that the causes may be seen as surprising, thus *relevance* and *consistency* are linked. Moreover, the similar notions of *stability* or *robustness* are also widely used to refer to the *continuity* and *consistency* criteria. We observe that some criteria play a role for different broader criteria, such as *completeness* which impacts both *generalizability* and *fidelity*. Indeed, the more complete an explanation is, the better it represents the original model which leads to better *generalizability* and *fidelity* but can worsen the *persuasibility* because of the added complexity of the explanation.

Overall, there are no disagreements in the criteria of an explanation and several main consensual criteria seem to emerge. Nevertheless, there is no consensus on the terminology, leading to confusions in the definitions of several terms as illustrated by the use of "consistency" and difficulties to identify "root" criteria i.e. criteria that cannot be divided into sub-criteria. We believe that Nauta et al. [42] managed to propose good candidates for such "root" criteria, striking a good balance between the number of criteria and their "rootness". Moreover, the profuse amount of criteria and the related confusions are easily observable on four reviews of the literature aimed at identifying and generalizing the propositions of other authors. Now that the criteria to evaluate the quality of an explanation have been discussed, we will explore the existing methods to quantify them.

Evaluating the criteria of an explanation

The identification of relevant, quantifiable and measurable criteria is critical to propose standardized benchmarks of an explanation. We have seen several candidates for criteria that were pro-

posed based on intuition, theories and studies from social sciences. These criteria are used to evaluate the quality of an explanation, but their relevance for such evaluation must be assessed. To this end, researchers have been working on methodologies to quantify these criteria as well as studying their relevance. Some criteria are subjective by nature, especially those relating to *persuasibility* thus requiring the intervention of humans to evaluate them. Consequently, evaluations are divided into objective evaluations and subjective or human-centered evaluations [41, 42, 96]. Doshi-Velez and Kim [67] proposed 3 categories of evaluations that differ in the type of task and whether humans are involved. *Functionally-grounded evaluation* requires no human experiments but uses proxy metrics to assess the explanation quality. This type of evaluation is purely objective and is ideal for classes of methods that have already been evaluated with human-centered evaluations or when a method is not yet mature. The main challenge for this type of evaluation is to determine the correct metrics that accurately represent a subjective criterion. *Human-grounded evaluation* involves human experiments to evaluate methods on simplified tasks, allowing for a bigger subject pool and thus less expenses. It is particularly useful to assess the relevance and contribution of each criterion in the evaluation of an explanation [41]. Finally, *application-grounded evaluation* also involves humans, but on real tasks. The subjects must be carefully selected to correspond to the target audience of the method e.g. when designing an explainability method for medical diagnosis, the evaluation should only involve doctors.

Although human-centered evaluations are highly relevant and may provide precious insights on the quality of an explanation, the cost and difficulty of finding participants is often prohibitive. Chromik and Schuessler discussed in further details the problems associated with the recruitment of participants for these user-studies [98]. The recruiting difficulty is likely to increase with the required level of participants' expertise. Therefore, objective evaluations are usually preferred by researchers in XAI because they are easy to manipulate and minimize cost [67, 98]. Some criteria can be evaluated with objective measures since they are naturally objective such as *compactness*, *continuity* or *consistency*. The challenge is determining accurate measures for subjective criteria i.e. create metrics that accurately approximate the evaluation of a human for a given criterion. These evaluation techniques are usually designed for specific types of explanations e.g. decision rules and feature importance methods do not function similarly and therefore the evaluation should be adapted. We will first present some evaluation methods linked to *generalizability* and *fidelity* as they are objective criteria, then we will discuss objective metrics that act as proxies of *persuasibility* criteria that are subjective by nature.

Yang et al. [40] argued that evaluation methods on *generalizability* are mostly focused on global explanations. Evaluating the *generalizability* of global explanations is equivalent to evaluating their *fidelity* to the original model. This evaluation method uses traditional metrics (e.g. accuracy, AUC or F1-score) to ensure that the predictions of the surrogate model are similar to the predictions of the original model. Nauta et al. [42] identified more than 25 papers that discuss this type of method to evaluate the *completeness* and also the *fidelity* of global explanations. This criterion should also be evaluated for different types of explanations. Concerning example-based methods, Nguyen and Martínez [99] introduced the non-representativeness and diversity metrics. Non-representativeness indicates how representative a set of examples is to explain a prediction. Diversity measures the difference between the examples, with the idea that the more diverse the set of examples is, the more generalizable the explanation will be. These metrics also evaluate *fidelity*. A high value of non-representativeness indicates that the examples poorly explain the prediction. It was argued that *continuity* and *consistency* (in the sense of Nauta et al. [42]) are equivalent to the notions of *stability* and *robustness*. Montavon et al. [95] quantified *continuity* as the strongest variation of the explanation function in the entire input domain. Alvarez-Melis and Jaakkola [94] discussed the same method, arguing that they are interested in evaluating the stability of the explanation only in the neighborhood of the input. To do so, they use the definition of local Lipschitz-continuity. Given two metric spaces $(X, \|\cdot\|_X)$ and $(Y, \|\cdot\|_Y)$, a function $f : X \to Y$ is locally Lipschitz if $\forall x, x_0 \in X, \exists \delta > 0, \exists L > 0, \|x - x_0\|_X < \delta \implies \left\|f(x) - f(x_0)\right\|_Y \leq L\|x - x_0\|_X$. Both methods propose a way to approximate a valid value for L which represents the *continuity* or

robustness of the explanation. This value is unit-less and the ideal value must be fixed depending on the application. Agarwal et al. [100] improved this method by taking into account the behavior of the original model. For instance, the behavior of the model may differ for two neighboring inputs if they are on either side of the decision boundary. To correctly explain the model's behavior, the explanations should therefore be different although the inputs explained are neighbors.

Sundararajan et al. [101] defined mathematical properties that should be observed in feature-based (or attribution-based) explanation methods; mainly the properties of sensitivity, implementation invariance and completeness. These properties ensure that the explanations accurately represent the original model thus increasing *generalizability* and ensuring *fidelity*. Beforehand, they introduced the notion of baseline that is crucial for the definition and evaluation of their properties. A baseline is the outcome of an AI system when given a neutral input. The definition of a neutral input depends on the nature of the input, the authors provided the example of a black image for object recognition tasks. Sensitivity imposes that when an input leads to a different outcome than the baseline and the input differs from the baseline in one feature, then the attribution (or weight) given to this feature should be non-zero. Implementation invariance states that two functionally equivalent models (i.e. their outputs are equal for all inputs) should lead to the same attributions independently of the implementation. Completeness (in [101]) verifies that the attributions add up to the difference between the output of the original function at the input x and the baseline x'. We note that sensitivity may have different definitions. Yeh et al. [102] defined sensitivity as a measure of the degree to which the explanation is affected by insignificant perturbations, linking back to the evaluation of *robustness*.

Although we observed that methods to evaluate *generalizability* can also be employed to evaluate *fidelity*, there exists many methods that only evaluate *faithfulness* or *correctness*. Controlled synthetic data check is used to evaluate the ability of an XAI method to produce truthful explanations in a synthetic task. It consists in creating a dataset where the discriminative features are known and controlled, allowing the production of ground-truth explanations. Oramas et al. [103] proposed the *an8Flower* synthetic dataset for visual explanations, consisting in images of flowers with zones of different colors depending on the class. The color and position of the zone along with its influence on the class are known and exploited to generate the ground-truths explanations. It is now possible to evaluate and compare several explanation methods on a same synthetic benchmark. Ribeiro et al. [49] proposed a similar approach. The model to explain and a set of interpretable models are trained on the same task to verify that the explainability method provides the same features as the interpretable models. Nauta et al. [42] warned that applying this evaluation method assumes that the black box has learned the intended reasoning, which is not always true. Therefore other evaluation methods must be used in conjunction to ensure that this assumption is reasonable. Several methods to evaluate *faithfulness* are directly inspired by *robustness* methods. While *robustness* checks that explanations are not sensitive to insignificant perturbations in the input, related *faithfulness* methods ensure that explanations are sensitive to significant perturbations in the input. These methods perturb the most important features that led to the prediction according to the explanation. This is usually done by setting the feature to the baseline such as a black pixel for an image, hence the name ablation analysis or "pixel-flipping" for image recognition tasks. The change in the output after applying these perturbations is measured to determine whether the causes given by the explanation have an actual impact on the prediction ([70, 75, 104]). Extensions of these methods perturb entire subsets of the features by setting each feature of the subset to the baseline value ([74, 105]). Alvarez and Melis [75] argued that feature removal is not always meaningful when applied directly on the input, they recommend removing the features at the model's level when possible. *Fidelity* related evaluations are well documented thanks to their objective nature and their proximity to robustness which is a well-established domain.

The objective evaluation of *persuasibility* related criteria is the most challenging as it includes inherently subjective criteria. The most commonly evaluated criterion is the size, *compactness* or complexity of an explanation. It is an objective criterion that depends on the nature of the explanation. Schwalbe and Finzel [69] listed several metrics to evaluate the architectural complexity

of explanations, namely the number of used input features for feature importance methods, the number of changed features (or sparsity) for counterfactual examples, the sparsity of linear models or the width and depth of decision trees. For feature-based methods, Bhatt et al. [74] evaluated the entropy of the distribution of the fractional contribution for each feature. The underlying idea is that if each feature had an equal contribution, the explanation would be too complex. Inversely, the simplest explanation would be concentrated on one feature. Nguyen and Martínez [99] proposed the effective complexity measure, which computes the minimum number of features that sufficiently explain the prediction. A low effective complexity means that some features have a small effect on the prediction and can be ignored, thus the number of features or *compactness* is minimized.

For other notions related to *persuasibility*, we mostly find the evaluations based on ground-truths discussed earlier, that are able to evaluate the relevance of the choice of features. Some methods evaluate the realism or feasibility of explanations by evaluating their distance to real-world examples. Heusel et al. [106] introduced the "Fréchet Inception Distance" which captures the similarity of generated images to real ones. Counterfactual explanations methods evaluate the degree of difficulty to achieve the counterfactual suggestions i.e. how attainable is the counterfactual example [107]. In summary, objective metrics developed to quantify subjective criteria are scarce and highly dependent on the nature of the XAI methods, with a particular focus on feature-based methods. Moreover, the relevance of these proxy metrics to evaluate subjective criteria is poorly evaluated and thus are not sufficient to assess the quality of an explanation in regards to subjective criteria. Freitas [108] deplored that too many papers evaluate the *persuasibility* or *comprehensibility* of a model solely on its complexity.

User-studies are ideal to evaluate the subjective criteria of an explanation, although they add a layer of complexity and cost to the evaluation of explanation methods. Indeed, a special attention must be paid to the methodology in order to get unbiased and relevant results. Chromik and Schuessler [98] presented a taxonomy of human subject evaluation in XAI. They identify three dimensions that play a role in the design of such studies: 1) the task of the study i.e. what is the intended goal of the explanation, what is evaluated and which information is presented; 2) the study design referring to design of the evaluation (e.g. qualitative, quantitative or mixed metrics) or the different treatments between subjects; and 3) the participants, which profiles to select, how many participants to recruit and how to efficiently recruit them. User-studies can evaluate three criteria according to Hoffman et al. [109] A test of satisfaction measures participants' self-reported satisfaction. It plays an important in gaining the user's trust but it is not necessarily correlated with their level of understanding [98]. To assess the level of understanding of the explained system, a test of comprehension is proposed. This test evaluates the mental model of a user about the explained system after receiving an explanation. In other words, it measures how well a human understands the functioning and capabilities of the explained AI. Finally, a test of performance evaluates the gain in performance of the human-AI system, as it is argued that a good explanation should improve the user's mental model and thus lead to an increase in performance.

These three types of test have been performed in the literature. A test of performance was designed by Huysmans et al. [110] where the goal was to test and compare the gain of performance of novice human subjects after being presented several interpretable models (e.g. decision trees, decision tables and textual descriptions). Subjects' accuracy, answer time and confidence were evaluated on different tasks. One of the main conclusion of this study is that *compactness* impacts these three measured criteria. Larger representations led to a decrease in confidence and accuracy as well as an increase in answer time. Förster et al. [43] proposed a human-grounded evaluation to compare the satisfaction level with different explanations. In this study, users were asked to match a leaf to one of four presented species of leaves associated with corresponding images. Then, two contrastive explanations were presented, with the form: "The leaf was classified as y_{fact} and not y_{foil}. In order to be classified as y_{foil}, the leaf would need to be <comparative><adjective> ... and <comparative><adjective>", where y_{fact} is the correct class and y_{foil} is the class predicted by the user if they guessed wrongly or the second most likely prediction if they guessed correctly. The

users were asked to choose the explanation they preferred. They could also answer that both explanations are unsuitable. Finally, users were asked the reasons for this choice, by choosing among several characteristics e.g. *long* or *short*, *general* or *concrete*... This study enables the comparison between similar methods and also the identification of characteristics that significantly impact the user satisfaction. The results of this study revealed that *concreteness* (the opposite of *generality*), *coherence* and *relevance* were decisive characteristics to choose the best explanation. These results differ from the literature in social sciences which usually states that general and short explanations are preferred. Förster et al. finally argued that the length or *compactness* of an explanation must be strategically chosen as an explanation that is too short may hinder the perceived *concreteness* thus having a negative impact on the user's appreciation. Conversely, the concentration of an explanation on few but relevant causes may increase the perceived *relevance* and consequently the overall satisfaction. Schraagen et al. [13] carried out a test of comprehension and a test of satisfaction with the aim of testing the relevance of the trust scale designed by Hoffman et al. [109] as well as evaluate the impact of different types of explanations on the user's satisfaction, trust and mental model. These types of explanations are causal, intentional and a mix of both. Causal explanations simply present the causes of a decision, intentional present the reasons leading to a decision and mixed present a combination of reasons and causes. The study presented decisions of an autonomous car in different situations along with an explanation for each decision. Participants were split into several groups and each group had different explanations, based on causes, goals or a mix of both. Satisfaction and trust were measured using the explanation satisfaction scale and trust scale from Hoffman et al. [109] as well as the trust scale by Jian et al. [111]. The test of comprehension or mental model accuracy was measured by two questionnaires developed by the authors. Users were asked to predict the behavior of the vehicle in one questionnaire while the second questionnaire evaluated the participant's knowledge about the vehicle's abilities. Ten situations were presented to the participants. After the third, sixth and tenth situation they were asked to fill out the trust scale of Hoffman et al. They filled the trust scale of Jian et al. after the third and tenth situation. Finally, the mental model questionnaires and the explanation satisfaction scale were filled after the ten situations. Participants were significantly more satisfied with intentional and mixed explanations than with causal ones. The trust scale of Hoffman et al. [109] was deemed valid and reliable. Although both scales contain a number of identical items, the authors noted opposite trends on the evolution of trust. They hypothesized that this phenomenon may be attributed to the order of the statements in the scales. The scale of Jian et al. starts with five negatively formulated statements while the scale of Hoffman et al. contains mostly positively formulated items. It was observed that the difference in the nature of explanations only have a short-end impact. Indeed, the differences between explanations disappeared after the second measurement of trust on the scale of Jian et al. . Mental models were not significantly affected by the type of explanation. The authors concluded by showing that causal explanations are consistently less trusted, unsatisfactory and lead to less effective predictive mental models than the other tested types of explanations. Mixed explanations led to the best functional understanding of the system and resulted in the least changes in trust over time.

Human subject evaluations allow scholars to test and improve theories about what constitutes a good explanation as well as compare the quality of different methods of explanations. Nevertheless, they have several limitations which hinders the generalizability of their results. There is no general and universal task that enables a fair comparison and evaluation of all the methods of explanations. Each task comes with different set of implicit expectations which impacts the metrics evaluated. Humans do not have the same requirements to trust the classification of a leaf compared to the decision process of an autonomous car. Such difference may reflect in the results and limit the comparison between studies. Similarly, the variability in the participants limit the generalizability of a study's results. Förster et al. [43] filtered participants based on the participant's perceived expertise in AI and botany. However participants did not share a common definition of what an expert is and this variation in expertise may affect the results. Huysmans et al. [110] stated that participants were not experts in the interpretable models, therefore the

results may significantly differ when the same study is presented to expert users. The empirical nature of user-studies implies imperfections and incompleteness in the study which also alters the generalizability of the results. Schraagen et al. [13] said that a limitation of their study was the way the situations were presented. They would have wanted to place participants in a driving simulator to fully understand the situation and better assess the explanations but this was not practically feasible. Huysmans et al. [110] noted that the excellent results obtained on a specific type of decision tables are not necessarily applicable to other types of decision tables. Likewise, the particular interpretable models presented to the users may be easier or harder to understand than usual because of the task. It is therefore difficult to generalize the obtained results without similar user-studies to provide additional insights.

The XAI community is working on identifying and measuring quantifiable criteria of an explanation that accurately describe the quality of an explanation in a given context. Some criteria can be quantified objectively, mostly regarding the mathematical properties of explanation such as *fidelity* or *generalizability*. Researchers are actively working on the design of evaluation methods that do not require the intervention of human subjects. However, most of the methods proposed are focusing on evaluating feature-based method and are not applicable to other XAI method. Other criteria are subjective by nature and require user studies to correctly evaluate them. Moreover, criteria are mostly derived from intuitions and theoretical studies from other domains which are not necessarily true. There is a clear need for more user studies that compare XAI methods on specific tasks, verify the relevance of criteria and identify new criteria of an explanation. The results from the existing user-studies show that contrary to theoretical studies, criteria do not have an ideal value e.g. an explanation should not necessarily be short to be good. Instead, the ideal values should be based on the task, the context and the user's needs. This observation indicates that the design of a general benchmark is near-impossible to make and not a desirable achievement. Finally, the terminology issue is also present in the design of evaluation of XAI methods with the use of the same term with different definitions (e.g. consistency) or different terms with a same meaning (e.g. fidelity, truthfulness and correctness).

3.1.4 Limitations

In this section, we reviewed the main problems faced by XAI and how scholars are solving them. The field of XAI provides techniques that explain AI systems in order to fulfill different goals e.g. justify the decisions of an AI system, design a responsible and ethical AI, increase trust, control or improve AI systems. The design of these techniques is particularly challenging because of the historical difficulty of understanding what constitutes a good explanation. As a result, researchers in XAI have been working on determining the components of an explanation and the criteria to evaluate the quality of an explanation. XAI methods should provide satisfaction to the user and increase trust in the system while respecting the principles of responsible AI. In our analysis of the literature, we identified unsolved problems and unexplored research tracks that motivated the contributions of this thesis.

The main issue that appears in every aspect of XAI is a general lack of consensus, which leads to the problems discussed in Section 3.1.1. The first aspect that is clearly affected by this lack of consensus is the terminology. Table 3.2 illustrates this phenomenon, with the proposal of 24 terms that have different but overlapping definitions. Overall, we observed identical terms that have different definitions and different terms that have similar or overlapping definitions. For instance, the terms explainability and interpretability are sometimes defined as synonyms, sometimes defined differently although they are among the most important terms in XAI. Moreover, authors do not systematically define terms used in their papers thus increasing the general confusion in the terminology. The lack of consensus for the terminology is observable in the many taxonomies proposed in the literature. Consequently, the identification of the components of an explanation and the criteria to evaluate the explanation's quality is particularly challenging.

The identification of the components of an explanation was discussed in Section 3.1.1. Despite a disagreement in the terminology, researchers seem to agree on three components: the

explanans, the explanandum and the explanatory relationship that link everything together. Figure 3.1 illustrates these components and their relationship. The choice of explanatory relationship or causal reasoning is poorly addressed in XAI techniques. The absence of causal reasoning implies that the explainee must infer it on their own hence hindering the quality of the explanation. Similarly, the context of the explainee is rarely taken into account in the terminology and design of explanation methods, although authors agree that it has an important influence on the quality of the explanation.

Evaluation strategies also suffer from the lack of consensus. Scholars seem to share the same view on the criteria that represent the explanation quality. Nevertheless, we observed in Section 3.1.3 that they struggle to find a common ground on the names, definitions and mathematical formulae of these criteria. Furthermore, it is clear that the quality of an explanation cannot entirely be evaluated with objective metrics (i.e. mathematical properties). Indeed, the social nature of an explanation implies that its quality is influenced by subjective criteria that varies depending on the explainee. Human subject evaluations enable the evaluation of these subjective criteria. Besides, they also verify the relevance of the identified criteria in the literature which may lead to the emergence of a consensus. Yet, several literature reviews on the evaluation of XAI methods noted the scarcity of human subject evaluations, mostly due to their cost and difficulty to setup. The conclusions of the existing user-studies coincide with the observations from social sciences. The quality of an explanation is highly dependent on the explainee and the context of the explanation. User-studies show that there is no ideal explanation but rather a set of choices and compromises to make in order to design the best explanation suited for a specific individual in a particular context for a given task. Nevertheless, some criteria, such as faithfulness, should always be met to ensure that the explanation follows the principles of Responsible AI.

As a consequence of the non-existence of an objectively perfect explanation, scholars design XAI techniques that correspond to their own idea of a perfect explanation. This results in a wide variety of explanation techniques that is difficult to navigate for newcomers but gives a lot of choice to find the best suited method. Although we observed a large focus on feature-based methods at the expense of other techniques. The community divides these XAI methods into two main categories. Post hoc methods are ideal to explain black-box algorithms while ante hoc methods exploit the interpretable nature of some AI algorithms to generate an explanation. Several authors are advocating for the use of ante hoc methods and therefore interpretable models. They argue that post hoc methods lack robustness and faithfulness which could lead to counterproductive results. Conversely, "traditional" interpretable models (e.g. linear models, decision trees or rules) generally achieve poorer performance than their opaque counterparts, especially when handling high dimensional data. Even in cases where performance are matched, the high dimensionality would result in an increase of complexity (i.e. an increase in the number of features or nodes), a loss of interpretability in the features and therefore a decrease of interpretability of the model. Neurosymbolic approaches are a response to this problem as they combine the interpretability of symbolic approaches with the qualities of machine learning models. In parallel, self-explainable models are being developed with the same goal of providing high performance and high interpretability.

From this review, we conclude that explanation methods alone cannot achieve the goals of XAI. Instead, we argue the design of an explainable system architecture, as depicted by the DARPA should be explored. This explainable intelligent system handles the life-cycle of the AI algorithm (e.g. training and inference parts) and the explanation task i.e. interact with the user to determine which explanation fits best and use some XAI methods to generate this explanation. To the best of our knowledge, such system does not exist. Although DARPA's XAI program [15] produced this architecture, the resulting contributions of this program did not focus on implementing such system.

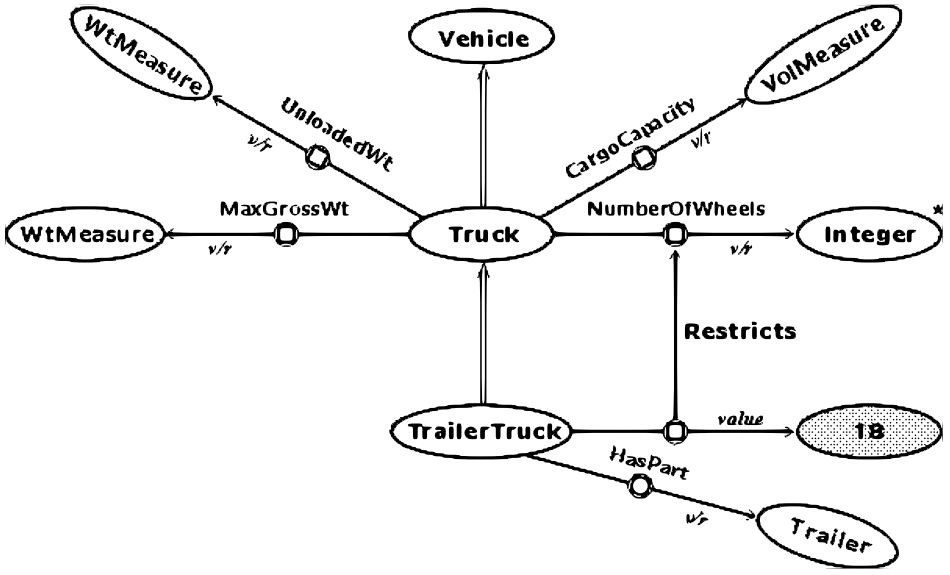


Figure 3.3: Example of a semantic network that describes a truck [114].

3.2 Background on ontologies

A symbolic AI system works by carrying out a series of logic-like reasoning steps over language-like representations [112]. Logical reasoning is the application of a set of rules (or logic) to infer or validate propositions based on existing knowledge. In other words, symbolic AI systems represent a problem with symbols that are comprehensible by humans, then apply some form of reasoning to deduce new facts or check the consistency of facts [113]. These symbols are concepts that are connected together through a set of relations. A good illustration of knowledge representation is a semantic network [114]. A semantic network is a graph where nodes are concepts and edges are the relationship between these concepts. Figure 3.3 displays an example of a semantic network that describes the concept of a truck. The nodes are concepts such as truck, vehicle or related measures while the edges are the relationships between these concepts e.g. a truck is a kind of vehicle, a truck has a certain cargo capacity. Another form of knowledge representation is the declaration of rules or clauses about symbols. For instance, the declaration `human(socrates).` specifies that `socrates` is human. Some form of logical reasoning is then applied on the symbolic representation to deduce new information or make a decision.

Modern or symbolic logic is an artificial symbolic language to provide a reasoning framework that is free from the constraints of human language [115]. The goal of logic is to infer a new statement based on a set of premises. For instance, if it is known that $A \implies B$ and the statement A is true, then we can infer that B is true. A large variety of such symbolic languages or logics have been developed to deal with different elements. They differ according to several criteria. One important criterion is the expressivity[4] of the logic i.e. the measure of what can be said (or expressed) in a logic [116]. Other criteria such as computational complexity, intuitiveness [116] or decidability [5] also guide the choice of logic. Indeed, logics that have a high expressive power are more likely to be undecidable, unintuitive or computationally complex compared to logics with less expressivity. First-order logic (also called predicate logic) is the most widely used logic in knowledge representation, but reasoning with this language is computationally expensive [118]. Consequently, some logics are built with fragments of the first-order logic that allows them to have sufficient expressivity for their application while reducing the reasoning complexity. Horn clauses for logic programming, or description logics for knowledge representation are examples of such logics applied to symbolic AI.

[4] Expressivity is also named expressiveness or expressive power.

[5] A logic T is decidable if there exists a method that permits to decide in each particular case whether a given sentence formulated in the symbolism of T can be proved by means of the devices available in T[117]

In this thesis, we intend to exploit a symbolic AI approach to design XAI solutions. We have discussed in Section 3.1.2 that ontologies have been identified as good candidates for this purpose. Hence, in this section we study ontologies. First, we introduce the notion of ontology and the different types of ontologies. Then, we discuss the description logics that are used by ontologies to formally represent knowledge and reason about it. Finally, we describe the OWL language and review its vocabulary that will be used in this work.

3.2.1 Ontologies and their applications

In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members). The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application. [119]. In other words, an ontology is a commonly agreed upon model of a domain of discourse that is specific and clear enough that it can be interpreted by a computer [120]. Uschold and Gruninger [121] identified three main uses for ontologies: communication, inter-operability and systems engineering. Ontologies can be used to provide a unifying framework within a community or organization to prevent conceptual and terminological confusions that typically arise when people communicate. Ontologies are also used to address the problem of inter-operability that occurs when users need to exchange data or use different software tools. An ontology acts as a unifying tool that standardizes these exchanges. Finally, ontologies are applied in systems engineering to assist the design and development of software systems and participate in their specification, reliability and reusability.

An ontology is used to describe a particular domain but it can be argued that all domains share the same generic concepts such as "objects" or "processes". It is thus possible to create more general ontologies that describe these generic concepts and the relations among them. These high level ontologies are called *foundational* or *upper ontologies* and represent very general concepts that are common across all domains [122]. They describe metaphysical and philosophical views of reality. Then, *core ontologies* specialize the upper ontologies to represent concepts that encompass hundreds of applications within the same field e.g. the Core Ontology for Multimedia Annotation (COMM) [123] represents any media object and is based on the DOLCE upper ontology. Finally, the most specific ontologies are *specific domain ontologies* and are built upon *core ontologies* to define concepts that are specific to one application. Figure 3.4 illustrate these three levels of ontologies. The design of *core* and *domain* ontologies is facilitated by Ontology Design Patterns (ODPs). Packages of ODPs are frequently used to solve commonly occurring modeling problems [120]. They follow the same hierarchy as ontologies, as generic ODPs are used to create more specialized patterns that are used themselves to design a specific domain ontology. This hierarchy can be observed with the Description and Situation (DnS) ODPs [124] that are extracted from the upper ontology DOLCE+DnS Ultra Lite . They are are extended to create more specific design patterns about multimedia in the COMM core ontology [123]. In turn, these specific patterns are employed to create specific domain ontologies in the domain of multimedia.

The development of formal ontologies that are machine-readable stems from the concept of Semantic Web. Its goal is to provide a common framework that allows data to be shared and reused across application, enterprise and community boundaries [125]. The stack in Figure 3.5 represents the several layers of technologies required to build the Semantic Web [126]. Each block represents a standard that is developed or is being developed by the W3C. Unique resource identifiers (URIs/IRIs) on the first layer depict each resource in the web e.g. a book, an author. The second and third layers (XML and RDF) provide languages and frameworks to connect these resources together e.g. Shakespeare is the author of Macbeth, the resources "Shakespeare" and "Macbeth" are linked with the relation "is the author of". The fourth and fifth layers (RDFS and OWL) propose more expressive solutions to represent connections between resources. Ontologies are part of the fifth layer and are applied to describe the sets of resources that share the same characteristics and the relations between different sets. For instance, an author is defined as a human in an ontology

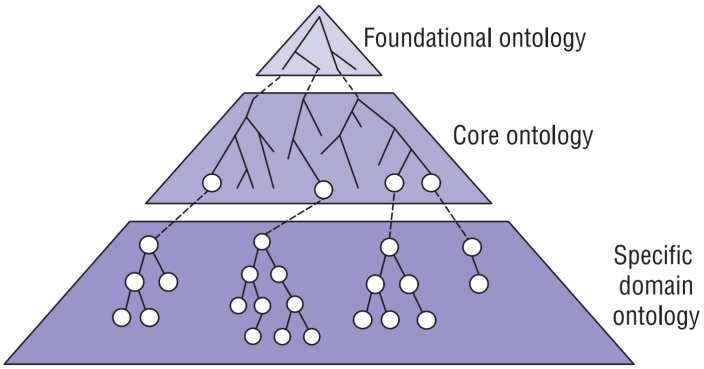


Figure 3.4: The three levels of ontologies [122]

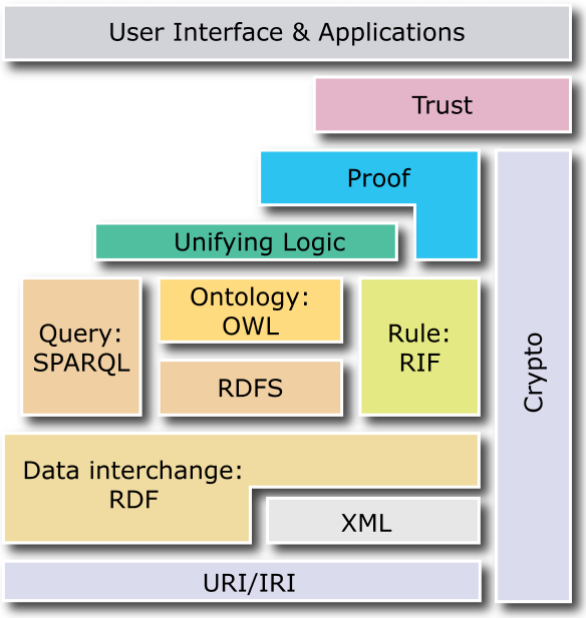


Figure 3.5: The Semantic Web Stack [126]

and has therefore a date and place of birth and many other relations with other types of resources that could not be expressed with the previous layers. The Web Ontology Language (OWL) was designed to create ontologies that is more expressive than RDF or RDFS while still compatible with them. We will further describe these technologies in Section 3.2.3.

Recently, major tech companies have started developing large knowledge graphs to connect resources found on the web. Similar to ontologies, knowledge graphs represent concepts and their relationships. The distinction between an ontology and a knowledge graph is ambiguous in the literature, due to their vague definitions [127]. Hence, ontologies and knowledge graphs share the same name and have the same structure. Some define a knowledge graph as an ontology with the data, which clashes with the definitions of an ontology proposed by the Semantic Web and Descriptions Logics. The DBpedia project [128] illustrates this ambiguity as they developed the DBpedia Ontology to serve as a schema for their knowledge base. Each resource described in the knowledge base is an instance of a class in their ontology. In this manuscript, we take the stance that a knowledge graph is a graph representation of an ontology. A same ontology can be represented in other forms such as a set of statements in description logics. Furthermore, the Web Ontology Language allows the transition from one type of representation to another.

This naming problem is caused by a wide adoption of ontologies in many domains outside of the Semantic Web. For instance, they are applied to medicine with the Gene Ontology [129], or finance with FIBO [130]. Projects such as DBpedia [128] or Wikidata [131] are large-scale ontolo-

gies that represent a wide variety of domains by extracting structured content from information available on the web, mainly from Wikipedia. These projects are then used for various applications such as natural language processing, question answering or knowledge extraction. Another famous ontology is WordNet [132] which represents more than 166 000 words with their definitions and semantic relations including synonymy, antonymy or meronymy. It enabled the design of the famous ImageNet dataset [133] that associates images to words of WordNet and led to major breakthroughs in computer vision and convolutional neural networks.

We have discussed the ability of ontologies to represent knowledge and data in a structured manner that is machine-readable. We also mentioned that symbolic AI techniques associate a knowledge representation with logical reasoning to make inferences. Ontologies employ a formal language that permits the application of a logical reasoner to infer about the represented knowledge. Description Logics are the family of logics employed to create ontologies and reason about them. In the next sections, we describe Description Logics and then the OWL language that was created by the W3C and uses the formalism of description logics.

3.2.2 Description logics

Description Logics (DLs) are a family of logics dedicated to representing the knowledge of an application domain in a structured and formal way [118]. A description logic describes *concepts* that denote sets of individuals and *roles* that are binary relationships between individuals. *Atomic concepts* and *atomic roles* are the basis of the knowledge representation. For instance, `Person` is an atomic concept and `hasChild` is an atomic role because they are not defined based on other concepts and roles. More complex descriptions of concepts and roles can be expressed with a description logic by using these atomic roles and concepts. The concept `Parent` can be defined as a person that has a child. This concept is formally defined as:

$$\texttt{Parent} \equiv \texttt{Person} \sqcap \exists \texttt{hasChild}.\top \tag{3.1}$$

where $\top$ refers to the universal concept. This definition means that the concept `Parent` is defined as a `Person` that is connected to another concept with the role `hasChild`.

A knowledge base designed with a description logic is composed of two elements: the TBox and the ABox [134]. The TBox (Terminology Box) contains all the axioms that constrain concepts of an application domain. For instance, the proposed definition of the concept `Parent` belongs to the TBox. The ABox (Assertions Box) corresponds to the data of the knowledge base and contains particular individuals and their properties. For example, the ABox contains the assertions: `Person(BOB)`, `hasChild(BOB, ALICE)`. These assertions state that the individual `BOB` is a `Person` and is related to `ALICE` with the `hasChild` role. The application of logical reasoning on a knowledge base declared with description logic allows the discovery of new facts about the individuals, the concepts and the roles of the knowledge base. In the previous example, we can infer that the individual `BOB` is also an instance of the concept `Parent`. Some particular description logics also have an RBox that is similar to the TBox as it contains all the axioms constraining the roles of an application domain [135].

The structure of a DL knowledge base resembles a database where the TBox corresponds to its schema and the ABox is compared the actual data that populates it. Yet, an important feature of DLs distinguishes them from database modeling languages: the *open-world assumption* (OWA) [134]. In a traditional database, the information is assumed to be complete, meaning that any statement that is true is also known to be true. This assumption is the *closed-world assumption* (CWA) and implies that statements that are not declared in the database are wrong. The *open-world assumption* is the opposite and considers that the absence of information only indicates a lack of knowledge. With this assumption, true statements are not necessarily known to be true and are therefore not necessarily declared in the ABox. For instance, the individual `BOB` was defined as having the child `ALICE`. With the CWA, `ALICE` would be considered an only child whereas with the OWA, it is unknown whether `ALICE` has siblings and this possibility is reflected in the inferences.

Notation	Meaning
$\mathcal{AL}$	Attributive Language. Base language that supports atomic concepts, atomic negation, concept intersections, universal restrictions, limited existential quantification
$\mathcal{C}$	Arbitrary concepts negation
$\mathcal{S}$	Equivalent to $\mathcal{ALC}$, with the addition of transitivity statements
$\mathcal{E}$	Full existential quantification
$\mathcal{U}$	Union of concepts
$\mathcal{F}$	Functional properties
$\mathcal{H}$	Role hierarchy
$\mathcal{R}$	Limited role axioms: reflexivity, irreflexivity, role disjointness
$\mathcal{O}$	Nominals
$\mathcal{N}$	Cardinality restrictions
$\mathcal{Q}$	Qualified cardinality restrictions

Table 3.3: Description Logics naming scheme [134, 136]

We have discussed DLs as a family of logics that share the same building blocks without mentioning the differences between each type of DL. Description logics are defined as extensions of the basic description language $\mathcal{AL}$ (for Attributive Language). This language only supports the definition of the basic syntax rules. Let C be a concept and R a role, the syntax of $\mathcal{AL}$ supports atomic concepts, the notions of universal and bottom concepts noted respectively $\top$ and $\bot$, the atomic negation $\neg$, the intersection $\sqcup$, the value restriction noted $\forall R.C$ and the limited existential quantification noted $\exists R.\top$ [134]. The expressivity of this language is limited as it cannot express the existential quantification with specific concepts, the union of concepts or the negation of arbitrary concepts. Consequently, extensions of this language have been introduced that each add new syntax rules. A letter corresponding to the new syntax rule is added to the language name to denote the resulting language. For instance, the letter $\mathcal{E}$ indicates the full existential quantification rule, the letter $\mathcal{U}$ denotes the union of concepts and the letter $\mathcal{C}$ refers to the negation of arbitrary concepts. Therefore, the language $\mathcal{ALUEC}$ is the basic attributive language extended with the union, the full existential quantification and the negation of arbitrary concepts. The complete nomenclature of description logics can be found in Table 3.3. We highlight that some letters may denote a combination of other letters, such as the language $\mathcal{S}$ that is equivalent to the language $\mathcal{ALC}$ with the addition of transitivity statements.

In the following, we will introduce the Web Ontology Language (OWL) that is used to define ontologies. There are several versions of OWL that each use a different DL [135]. The OWL 1 Lite Standard uses $\mathcal{SHIF}$ which is obtained from $\mathcal{ALC}$ by adding an RBox and thus the possibility to express constraints on roles. The OWL 1 DL standard is more expressive than the latter as it uses $\mathcal{SHOIN}$ that is obtained from $\mathcal{SHIF}$ by adding support for nominal concepts and unqualified number restrictions. Finally, the OWL 2 DL standard is the most expressive, it uses $\mathcal{SROIQ}$ which is obtained from $\mathcal{SHOIN}$ and adds even more possibilities on the definitions of roles as well as qualified number restrictions.

3.2.3 The Web Ontology Language (OWL)

The Web Ontology Language is a language that is based on Description Logics (see Section 3.2.2) to define an ontology i.e. concepts and their roles. As stated by the W3C [137], OWL is an extension of RDF and RDFS technologies that are present in lower layers of the Semantic Web stack in Figure 3.5. The Resource Description Framework (RDF) is intended to provide a metadata data model to the Web [126]. It uses triples as its basic unit of information. A triple is composed of a *subject* (s), a *predicate* (p) and an *object* (o) and is noted `(s,p,o)`. In the vocabulary of DL, the subject and object are concepts while the predicate is a role. For example, the triple `(BOB, hasChild, ALICE)` represents the fact that Alice is the child of Bob and corresponds to the assertion `hasChild(BOB,`

ALICE) expressed with the formalism of DL. A set of triples can be represented as a directed labeled graph where resources are the nodes and predicates represent the labeled edges that connect the subject node to the object node. We note that the terminology of RDF, RDFS and OWL differs from the one used in description logics. Concepts are called *classes* while roles are called *properties*. Furthermore, a predicate of a triple is necessarily a property (or role).

Any resource or predicate corresponds to a Unique Resource Identifier (URI). For instance, the resources Bob and Alice may have the following URIs: <http://www.example.com/BobFamily#Bob>, <http://www.example.com/BobFamily#Alice>. They both share the same base URI that is <http://www.example.com/BobFamily#>, that we call a namespace. To shorten and make RDF graphs easier to read, it is possible to declare a prefix to refer to this namespace e.g. the prefix bobf refers to the previous base URI. With this prefix, any resource with this base URI can be written as bobf:Bob. The standard namespace of RDF[6] is shortened by the prefix rdf before the name of the resource being denoted.

We observe that the resources in the RDF namespace enable the possibility to define concepts (i.e. sets of individuals) and roles. However, RDF alone is not expressive enough and only provides elementary typing abilities with rdf:type and rdf:Property [126]. According to the W3C [138], the RDF Schema (RDFS) is a semantic extension of RDF that adds the ability to further describe *classes* as well as *properties*. It adds the expressivity to describe hierarchies of classes and properties which was lacking in RDF. It also enables the characterization of a property by adding the notions of *domain* and *range*. For the triple (s,p,o), the *domain* of the property p refers to the class of s while the *range* of p is the class of o. It imposes that the subject and object of a triple are respective instances of the *domain* and *range* classes defined by the predicate. OWL further extends RDFS by adding the ability to express classes with a logical combination of other classes or as enumerations of specified objects. It also has the capacity to give certain characteristics to a property such as transitivity, symmetry, functionality or being the inverse of another property [139]. As we discussed in Section 3.2.1, there are several versions of OWL with the least expressive being the OWL 1 Lite that uses the DL $\mathcal{SHIF}$ and the most expressive being the OWL 2 DL that uses $\mathcal{SROIQ}$. The increase of expressivity from OWL 1 to OWL 2 is intended to address shortcomings identified by ontologist after years of experience on OWL 1 [140].

In this thesis, we use OWL 2 to design our ontologies. The structural specification of OWL 2 is described in [141] along with a functional-style syntax that separates the essential features of the language from issues related any particular syntax. Still, OWL 2 is part of the Semantic Web stack and must remain compatible with the other technologies. Therefore, they also propose a mapping from OWL 2 to RDF (given in [142]) which enables to represent OWL 2 ontologies as RDF triples and thus as a graph. We will briefly review the syntax and vocabulary of OWL ontologies that is used in our contributions. We first describe the vocabulary related to the definition of the TBox of an OWL ontology. Classes represent a set of individuals and are equivalent to the term *concept* in DLs. The universal class is noted owl:Thing while the bottom class is noted owl:Nothing. Properties represent the relationship between pairs of entities in an ontology and are equivalent to *roles* in DLs. Two types of properties are defined in OWL, object properties and data properties. *Object properties* connect pairs of individuals while *data properties* connect individuals with literals. A *literal* is a data value such as strings, integers or dates e.g. the string "Bob" is a literal while the individual Bob is an individual of the ontology. Several types of axioms to describe classes and properties can be expressed in OWL. The following axiom types will be used in the manuscript.

Subclasses A class can be defined as a subclass of a parent class, meaning that all individuals that belong to class B also belong to the parent class A. It corresponds to the following DL statement:

$$\underbrace{\mathtt{B} \sqsubseteq \mathtt{A}}_{\text{B subclass of A}} \iff \forall x (x \in B \implies x \in A) \tag{3.2}$$

Subproperties Like subclasses, a property can be defined as a subproperty of another property,

[6] https://www.w3.org/1999/02/22-rdf-syntax-ns#

meaning that if an individual x is connected to an individual y by a property $\mathtt{p}_2$ and $\mathtt{p}_2$ is a subproperty of $\mathtt{p}_1$, then x is also connected to y by the property $\mathtt{p}_1$. It corresponds to the following DL statement:

$$\underbrace{\mathtt{p}_1 \sqsubseteq \mathtt{p}_2}_{\mathtt{p}_2 \text{ subproperty of } \mathtt{p}_1} \iff \forall (x,y)\big((x,y) \in \mathtt{p}_2 \implies (x,y) \in \mathtt{p}_1\big) \tag{3.3}$$

Property domain and range The *domain* and *range* of a property have a similar definition to their RDF equivalent. The property domain is the class such that if an individual x is connected to another individual by a property p, x is an instance of the domain of p. It corresponds to this statement, where D is the domain of the property p:

$$\underbrace{\exists p.\top \sqsubseteq D}_{\text{D is the domain of } p} \iff \forall (x,y)\big((x,y) \in p \implies x \in D\big) \tag{3.4}$$

Likewise, the property range is the class such that if some individual is connected by a property p to the individual y, y is an instance of the range of p. It corresponds to this statement, where R is the range of the property p:

$$\underbrace{\top \sqsubseteq \forall p.R}_{\text{R is the range of } p} \iff \forall (x,y)\big((x,y) \in p \implies y \in R\big) \tag{3.5}$$

Functional property A *functional* property states that for each individual x, there can be at most one distinct individual y such that x is connected by the functional object property to y. It corresponds to this statement:

$$\underbrace{\top \sqsubseteq \leq 1 p.\top}_{p \text{ is functional}} \iff \forall x, \big|\{y,\ (x,y) \in p\}\big| \leq 1 \tag{3.6}$$

In other words, functional properties correspond to unique features of an individual e.g. the object property `hasBirthPlace` that connects a person to a location is functional, since a person can only have one birth place.

The syntax of OWL 2 allows the creation of *class expressions* also called *descriptions*. These class expressions represent sets of individuals by specifying conditions on the individuals' properties. There are several types of conditions that can be applied to design a class expression. Intersection, union or complement of classes are standard connectives and are one category of condition. Another category is property restrictions and cardinality restrictions. A property restriction ensures that individuals represented by the class expression are connected to other individuals with some property.

- Existential quantification is a property restriction that imposes that there exists a connection between the individuals represented by the class expression to another individual with a particular property. For instance, the class `Parent` can also be written as a person that has at least one child. It is possible to define the set of individuals that have at least one child by creating a class expression with an existential quantification on the property `hasChild`. This class is noted `ObjectSomeValuesFrom(hasChild, Person)`.
- Individual value restrictions ensure that individuals represented by the class expression are connected to a specific individual by a specified property.
- Cardinality restrictions enforce that individuals of the class expression are connected to at least, at most or exactly n different individuals with the same property. An equivalent definition of the same set of individuals can be expressed with a minimum cardinality restriction, imposing that the individuals have at least one connection to another individual with the object property `hasChild`.

Thanks to these class expressions, it is possible to describe the class `Parent` in OWL 2 as the intersection of a person that has a child:

$$\texttt{Parent} \equiv \texttt{ObjectIntersectionOf(Person, ObjectSomeValuesFrom(hasChild, Person))} \tag{3.7}$$

The expression `ObjectIntersectionOf` represents the intersection between class expressions and `ObjectSomeValuesFrom` refers to the existential quantification between an object property and a class expression. Here, the existential quantification represents the set of individuals that have some connections between them and instances of the class `Person` with the object property `hasChild`.

The ABox in OWL 2 exploits the TBox to represent particular individuals. Facts about an individual are stated with *assertions*. An individual can be defined as belonging to a certain class with the `ClassAssertion`. Individuals are connected together with several types of property assertions. The assertions `ObjectPropertyAssertion` and `DataPropertyAssertion` connect an individual to another individual or literal, with an object or data property defined in the TBox. Conversely, the assertions `NegativeObjectPropertyAssertion` and `NegativeDataPropertyAssertion` state that an individual is not connected to another individual or literal with a given object or data property.

We mentioned earlier that there are multiple possibles syntaxes for OWL 2. The syntax used in this section is the functional syntax but it may result in expressions that are complex and hard to read for a human. The W3C provides a mapping of OWL 2 ontologies into RDF graphs [141] which enables a graph of OWL 2 ontologies as it uses the triples syntax that we introduced in our discussion on RDF. There is also the Manchester syntax that proposes simpler expressions that look like sentences [143]. In the Manchester syntax, property restrictions are expressed with keywords. For instance, the existential quantification is noted with the word `some`, while minimum or maximum cardinality restriction are noted with `min` or `max`. The class expression to define the class `Parent` is written `Person that hasChild some Person`. The keyword `that` represents the conjunction or intersection between a class and a class expression.

3.2.4 Discussion

This section provides technical knowledge about ontologies to help the reader understand the remainder of the thesis. We have reviewed what an ontology is and how it is related to symbolic AI. We have presented important notions of Description Logics and then introduced the Web Ontology Language (OWL) that is heavily used to develop ontologies. Thanks to its recommendation status from the W3C, OWL benefits from a large suite of tools to develop and use an ontology. We mainly used the Protégé editor to develop our ontologies and the Owlready2 Python package as an interface between ontologies and our contributions. The Protégé editor [144] was developed by the Stanford University as a free open-source platform to develop and manage ontologies. It uses the Manchester syntax to design an ontology and proposes a set of logical reasoners that can be run directly from Protégé. The Python package Owlready2 [145] also allows the development of ontologies and the execution of logical reasoners, directly with code instead of a graphical interface.

The access to a logical reasoner is important as it makes inferences and checks the consistency of an ontology. Indeed, ontologies are usually very complex and it happens that some elements in the ABox or TBox provoke an inconsistency i.e. a fact and its opposite are both inferred as true. Moreover, we have discussed that DLs use the open-world assumption which further complicates the reasoning task as unknown facts are not considered false. Consequently, the reasoner needs to consider different hypotheses to infer new facts or check the consistency. Hence, designing a complex ontology that is consistent and accurately reflects the intentions of the designer is a difficult task. Despite the existence of solutions such as upper and core ontologies or ontology design patterns to facilitate this task, we will observe later in the thesis (see Section 6.1.2) that understanding the inferences and debugging an ontology is still an open problem.

Chapter 4

The XAI terminology

Contents

4.1 Literature review 58
4.1.1 Interpretability and transparency 58
4.1.2 Explainability and explanations 61
4.1.3 Responsible AI terms 62
4.2 A terminology for a contextualized XAI 64
4.2.1 System terminology 65
4.2.2 Explanation terminology 70
4.2.3 Formalization of XAI explanations 72
4.3 Conclusion 73

The recent interest in the design of explainable AI algorithms led to a surge in research papers in this domain. In Chapter 3, we reviewed the current state of the literature of XAI and identified a lack of consensus regarding the XAI terminology [28, 29, 63]. Particularly, several terms specific to XAI are commonly used but rarely defined. Furthermore, the proposed definitions for these technical terms are usually ambiguous because they use other terms that are not defined themselves. In addition, scholars do not agree on the meaning of some terms which results in contradictory definitions. The absence of consensus hinders the research in XAI as there is no agreement on how to make a system explainable or interpretable as well as the criteria for a good explanation. Moreover, it renders the XAI domain less accessible to newcomers and prevents good communication between members of the community.

First, we clarify the meaning of AI algorithm, AI system and model that are used throughout the thesis.

AI algorithm An AI algorithm is a set of instructions that tells the computer how to operate or learn to operate. In machine learning, the AI algorithm is equivalent to the training algorithm that finds the best parameters of a model for a specific task. Outside of machine learning, the AI algorithm is the set of instructions that directly make a decision based on an input.

Model We use the term model to refer to a machine learning model. A model is a mathematical function that contains a set of parameters that need to be optimized to carry out a specific task. The optimization of a model's parameters is done by an AI algorithm.

AI system AI system refers to a system that makes a decision based on a set of inputs. The AI system encompasses the input, the output or decision as well as the decision process to make the decision. This decision process is an AI algorithm or a model in the case of machine learning.

The notions of AI algorithm and model are quite similar. To avoid any ambiguity, model is only employed to describe a machine learning model while AI algorithm is used to refer to the decision process. To summarize these notions, an AI system is represented as $y = f(x)$ i.e. it is the set x, y, f where x is the input, y is the decision that was made using the function f. In the context of machine learning, f is called as model whereas in a more general context, f is called the AI algorithm. AI algorithm is never used to refer to a training algorithm in machine learning. We note that the literature on XAI is focused on explaining machine learning models, thus the term model is heavily used in the literature review while we prefer the terms AI systems or AI algorithms to remain generic about the type of AI.

In this chapter, we propose an unambiguous terminology that defines the recurrent terms specific to XAI. This terminology is based on a survey of the literature that aims to identify these terms and their associated definitions. The definitions are designed to be compatible with a majority of the definitions encountered in the literature. Finally, an ontology is introduced that describes the relationship between the concepts of the terminology and the ontology design pattern to define explanations introduced in [65] is instantiated to describe an explanation in the context of XAI.

4.1 Literature review

We conducted a literature review to identify the recurrent and important terms specific to XAI that are subject to disagreements and study their definitions. This review complements our survey in Chapter 3 in which we introduced several notions specific to XAI without exploring their definitions. For instance, the terms *ante hoc*, *transparent* model and *interpretable* model are closely linked but their actual definition is ambiguous. Therefore, the uses and the meaning of these terms by the XAI community is studied to determine their commonalities and differences and thus propose a coherent and unambiguous terminology.

We have searched the literature related to XAI, using the keywords "XAI", "terminology", "taxonomy", "survey", "review", "explainability" and "interpretability" in Google Scholar; looking for surveys, taxonomies and reviews that propose definitions of XAI terms. This survey along with the literature study conducted in Chapter 3 give a good overview of the terminology in XAI. In addition, we also reviewed papers that propose explainability methods to refine our definitions and make them compatible with a majority of these methods.

The most important terms to define are *interpretability* and *explainability*. Indeed, *explainable* is in the name of the field, while *interpretability* is still commonly used to refer to XAI as illustrated in Figure 4.1. Therefore, we first study the definitions and uses of the term *interpretability* and the related concepts (e.g. transparency, opacity...). Then, the meanings of *explainability* and *explanation* are analyzed along with the concepts used to categorize an explanation as described in Section 3.1.2. Finally, notions related to Responsible AI are reviewed as they are commonly goals that explainability methods aim to achieve.

4.1.1 Interpretability and transparency

Interpretability is a term commonly used by the community to describe AI systems that are easily understandable [29]. The implicit definition that some scholars employ is that an interpretable model is a model that can be understood by most users [34]. Some authors define interpretability as "the ability to explain or present in understandable terms to a human" [28, 67, 146]. However, we argue that this definition is very broad and the use of the term *explain* makes it confusing with the notion of *explainability*. Adadi and Berrada [29] proposed a more thorough definition that does not employ *explain*: "An interpretable system is a system where a user cannot only see but also study and understand how inputs are mathematically mapped to outputs". Gilpin et al. [93] noted that "the goal of interpretability is to describe the internals of a system in a way that is understandable to humans", aligning with the previous definition. They also add that "the success of this goal is tied to cognition, knowledge and biases of the user". Likewise, Calegari et al. [77]

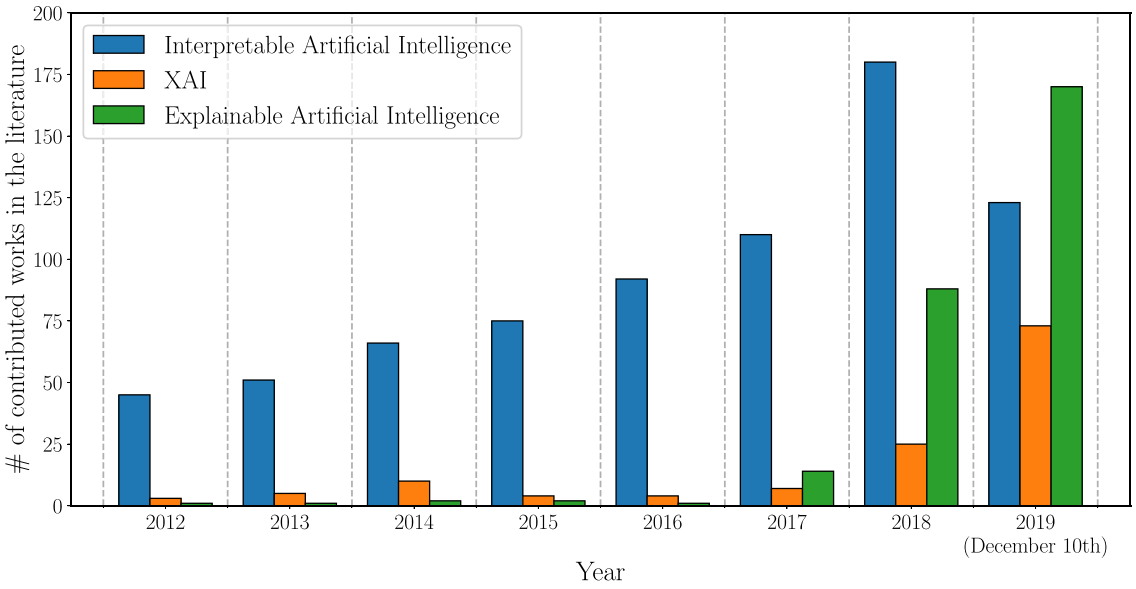


Figure 4.1: Evolution of the number of total publications whose title, abstract and/or keywords contained the terms in the legend [28].

stated that "interpretability refers to the cognitive effort required by human observers to assign a meaning to the way the algorithm works, or motivate the outcome it produces". From these definitions, two criteria of interpretability emerge. One criterion concerns the visibility of the system's internals i.e. the mathematical function that maps the input to the output. The other criterion is the cognitive effort required to understand the system's internals.

The cognitive effort required to understand the functioning of an AI system depends on the user [33, 49, 77] and the complexity[1] of the algorithm [45, 49, 63]. Although the impact of the user's expertise on the required cognitive effort cannot be objectively evaluated, the complexity of the algorithm only depends on its functioning. Therefore, scholars have proposed measures of complexity for a myriad of algorithms [45, 49, 93, 147, 148]. For instance, Wu et al. [148] compute the average decision path length as the complexity of a decision tree, while Ribeiro et al. [49] use the number of non-zero weights of a linear model to measure its complexity. Consequently, Lipton [63] stated that it is meaningless to qualify any model as intrinsically interpretable since interpretability depends on the complexity of a model. Thus, a decision tree with a large average decision path length has a high complexity and is unlikely to be interpretable for most users. Yet, each element of the decision tree is understandable as it consists in a simple rule. This leads us to transparency, the second criterion of interpretability.

Transparency is a controversial term, its meaning varies greatly. For some authors, transparency is closely tied to interpretability. Indeed, Lipton [63] observed that certain papers qualify understandable models as transparent while incomprehensible models are called opaque or black-box models. According to the definitions of interpretability studied above, an understandable model is an interpretable model. Thus, transparency and interpretability seem to depict the same notion. Based on this meaning, three levels of transparency that are all present in an interpretable model have been identified by scholars [28, 63, 147, 148]:

Decomposability Transparency at the level of individual components is called decomposability. Arrieta et al. [28] defined it as "the ability to explain each of the parts of a model (input, parameter and calculation)". Futia and Vetrò [147] proposed a similar that requires each part of the model to be interpretable instead of explainable; illustrating once more the disagreements between explainability and interpretability. For instance, an explanation for the the

[1] The notion of complexity used here, in the context of XAI, is not to be confused with the distinct notion of algorithmic complexity.

parameters of a linear model is that they represent strengths of association between each feature and the label. Several methods implicitly exploit decomposability to create interpretable models e.g. the SENN [75] and LIME methods [49] ensure that the inputs of their model are interpretable.

Algorithmic transparency Transparency at the level of the learning algorithm is called algorithmic transparency. It is defined as "the ability of the user to understand the process followed by the model to produce any given output from its input data" [28]. Lipton [63] illustrated this notion with the case of linear models, for which a user may understand the shape of the error surface. Inversely, the loss functions used by deep architectures may be difficult to understand rendering them not transparent. A slightly different definition is provided by Futia and Vetrò [147]. They define algorithmic transparency as a way to guarantee that a model behaves in an expected way thanks to the learning algorithm. A linear model converges to a unique solution whereas there is no guarantee that a neural network works in the same way on new problems because of the non-deterministic nature of the training algorithm (the solution found by a neural network depends on the initial weights, random seed and the order in which the training data is given).

Simulatability Transparency at the level of the entire model is called *simulatability* [28, 63, 147, 148]. A model is considered simulatable when given the input data and the parameters of the model, a human is able to step through every calculation required to produce a prediction in a reasonable time. The link between simulatability and complexity is further discussed. Lipton [63] mentioned tradeoffs between model size and computation. Indeed, the computation for a decision tree is easily carried out yet the size of these models may grow quite large. Similarly, Arrieta et al. [28] noted how a decision tree with a large amount of rules may not be considered simulatable whereas a single perceptron neural network may be seen as simulatable. Two subtypes of simulatability are consequently defined by Lipton [63]. One refers to the complexity of every computation while the other refers to the size of the model. We argue that the latter corresponds to the notion of complexity presented before.

Other authors place interpretability and transparency at different levels of granularity. Guidotti et al. [45] described a transparent box design as a system that learns a locally or globally interpretable predictor for which there exists an explanator. This design resembles the explainable intelligent system introduced in [15] and places transparency at the scale of the system while interpretability is at the scale of the model. Inversely, Arya et al. [34] stated that "a directly interpretable model is one that by its intrinsic transparent nature is understandable by most consumers". Therefore, interpretability is at a larger scale than transparency i.e. transparency is a component of interpretability.

A requirement for all these definitions of transparency is that information about the functioning of the model is available. Beaudouin et al. [146] proposed a definition of transparency based on a dictionary definition and a document from the OECD [149]. For them, transparency refers to "making information about the inner workings of the algorithm available for scrutiny, including how an AI system is developed, trained and deployed". Furthermore, they separated interpretability and transparency by stating that "transparency does not necessarily mean that the underlying information is easily comprehensible to humans".

From these definitions of interpretability, transparency and related notions, we gather that interpretability is a quality of an AI system that makes it understandable by most users. An AI system is understandable by a user it requires a reasonable cognitive effort to analyze it. Several factors impact the required cognitive effort such as the complexity, simulatability, decomposability and algorithmic transparency of the AI system. The difference between the notions of interpretability and transparency is unclear. For some authors, these terms are synonyms while other scholars give different definitions that do not reach a consensus. This ambiguity also comes from the opposite notion of opaque or black-box models. Opacity and transparency are antonyms in the dictionary but the XAI community qualifies non interpretable models as black-box or opaque models, adding

to the ambiguity between transparency and interpretability. Finally, there is also an ambiguity between the notions of explainability and interpretability. In the next section, we explore the existing definitions of explainability.

4.1.2 Explainability and explanations

Explainability is the main notion in the field of XAI but we have seen in Chapter 3 that the definitions of explainability and explanation are still in discussion. In this previous chapter, we observed that some scholars use explainability and interpretability interchangeably while others explicitly differentiate these terms. We prefer to use a more straightforward definition that is directly deduced from the construction of the word. The suffix "-ability" indicates that explainability is the ability to explain. In other words, an explainable system is a system that is able to explain itself. This brings back to the definition of an explanation.

Explaining something is equivalent to answering questions that a user may ask with the intent of understanding what they are observing [37, 68, 150, 151]. The aim is to provide relevant information so that the user can reason on their own about how a model works or why a model made a specific decision. Therefore, there is a notion of interaction intrinsic to an explanation [147, 151]. Calegari et al. [77] define an explanation as "an activity aimed at making the relevant details of an object clear or easy to understand to some observer". Arrieta et al. [28] and Guidotti et al. [45] share the same definition: "an explanation is an interface between humans and a decision maker that is, at the same time, both an accurate proxy of the decision maker and comprehensible to humans". From these definitions, an explanation is an interactive process that presents relevant pieces of information to make an AI system and/or its decisions comprehensible [49].

The relation between interpretability and explainability is that both make the functioning of a system understandable or comprehensible to its users. An explanation is an interactive process where the goal is to provide relevant details connected together by some form of reasoning (see Section 3.1.1) to answer the questions of a user. Interpretability is a property of a system, all the information is available to the user but there is no intervention from an agent to answer the user's questions. Thus, the main notable difference is that an explanation is an active process whereas interpretability is a property of a system that does not require any action to be understandable for the users.

Despite the difficulties to define an explanation, the XAI community agrees on several categories of explanations. We focused on two categories: global/local explanations and post hoc/ante hoc explanations. Other categories have been identified in the literature, such as the duality model-specific and model-agnostic methods (see Section 3.1.2) as well as static and interactive explanations [34] for which there is no debate regarding their name and definitions. We briefly introduced the definitions of these categories in Section 3.1.2 but did not further analyze the nuances in the definitions provided by the community.

Global/Local explainability The definitions of global and local explainability make consensus in the literature. However, since some scholars use interpretability and explainability as synonyms, they define global/local interpretability instead. For instance, Guidotti et al. [45] and Adadi and Berrada [29] state that "global interpretability facilitates the understanding of the whole logic of a model". Similar definitions are given in [34, 146, 152]. Hoffman et al. [151] describe global explainability a the explanation of "how the conceptual categories and mechanisms are derivable from instances and their attributes". Then, local explainability refers to explaining a single prediction made by an AI system. This idea is accepted in the community [29, 34, 67, 147]. However, Beaudouin et al. [146] consider local explainability as synonym of post hoc explainability which is not a meaning that we encountered in the rest of the literature. Although it is true that most post hoc explanations focus on explaining a single prediction, there are examples of global post hoc explanations, for instance ProfWeight [153].

Post hoc/ante hoc explainability The meaning of post hoc explainability is generally agreed in

the literature. According to Futia and Vetrò [147], post hoc explanations "do not seek to reveal how a model works, but they are focused on how it behaved and why". Lipton [63] states that "these interpretations might explain predictions without elucidating the mechanisms by which models work". Guidotti et al. [45] qualify them of "reverse-engineering approaches". These definitions show that post hoc explanations do not exploit the mechanisms of the AI system but manipulate it to understand its behavior. Arya et al. [34] define post hoc explanation methods as "auxiliary methods to explain a model after it is trained". These methods are valuable when the internal logic of the system is not available or too complex for most users to understand.

In opposition to post hoc explainability, scholars describe a type of explainability that exploits the interpretable nature (i.e. when the internal logic of an AI system is both available and understandable by most users) of the system to explain [28, 34, 77]. Nevertheless, there is no consensus on the term associated to this concept. We chose ante hoc as it seems the most coherent with post hoc[2] and is already employed by scholars [44, 46]. Scholars sometimes name this type of explanation as "directly interpretable" or "transparent model" [28, 32, 34, 44, 69]. Calegari et al. [77] call it explainability by design: "methods in this category aim at creating interpretable or explainable intelligent systems by construction". We observe that it is unclear whether these definitions describe a model or an explanation technique because they are qualified as opposed to post hoc explanations but seem to refer to models instead of explainability methods. Still, the underlying idea remains the same for all these definitions i.e. explain by exploiting the interpretable internal mechanisms of the system.

In summary, explainability is the ability of a system to generate explanations comprehensible by most users. Explanations are an interactive process between the system and the user with the goal of providing relevant details connected with some form of reasoning to facilitate comprehension. There are several categories of explanations which mostly focus on the scope (e.g. global/local explanations) and the method (e.g. post hoc/ante hoc and model specific/agnostic). We noted a difference in the nature of explainability and interpretability, the former is an active process conducted by the AI system while the latter is a passive property of such system.

4.1.3 Responsible AI terms

Responsible AI represents a set of principles that guide the design of an AI system. We reviewed these principles in Chapter 2 and analyzed how they were included and evaluated in Chapter 3. In this analysis, we observed many disagreements regarding the names and definitions of concepts related to these principles. Notably, the terms reliability, confidence, consistency, robustness, fidelity, truthfulness and correctness regularly appear in the literature with different meanings. The underlying notions are consensual but the terms attributed to each notion are different.

An important notion in XAI is how accurate an explanation is with regards to the AI system being explained. An explanation may be convincing but completely dissociated from the actual behavior of a system. The terms fidelity, correctness, faithfulness, soundness or truthfulness all relate to this idea that the explanation should be faithful to the AI system being explained [29, 40, 41, 42, 43, 63, 67, 90]. Nauta et al. [42] define this notion as a measure of the descriptive accuracy of an explanation. This notion is further discussed in Section 3.1.2.

A set of concepts related to the behavior of models is commonly discussed in the literature. Terms such as confidence and reliability are often employed to refer to the assurance that a model is providing the correct answer and behaves in an expected manner [28, 77]. Doshi-Velez and Kim [67] define reliability and robustness as properties that "ascertain whether algorithms reach certain levels of performance in the face of parameter or input variation". Guidotti et al. [45] propose a similar definition of reliability and robustness. Arrieta et al. [28] define confidence as "a

[2] Post hoc meaning "after this" and ante hoc meaning "before this" in latin [154].

generalization of robustness and stability" and state that "confidence should always be assessed on a model in which reliability is expected" thus associating confidence to reliability. There were few definitions for robustness and stability explicitly given in the reviewed papers. Still, they are common terms in AI and the authors of the studied articles employ consensual definitions. A definition of robustness in software systems is "the degree to which a system or component can function correctly in the presence of invalid inputs or stressful environmental conditions" [155]. In the context of AI algorithms, it is the ability of an algorithm to behave as expected in the presence of unexpected or erroneous inputs. The term stability, also called sensitivity, refers to how the behavior and outputs of a model remain stable when data is perturbed [29]. A stable learning algorithm is an algorithm for which the learned solution does not change much with small changes in the training set [156]. Therefore, robustness is about the ability of an AI system to behave in an expected way after the training step when inputs are perturbed, while stability is the ability of a learning algorithm to behave in an expected way when the training data is perturbed. Confidence and reliability are two terms that relate to the assurance that an AI system behaves in an expected manner which necessitates stability and robustness.

Confidence is also used with a different meaning in other contexts. It is defined as an estimation of the correctness of a prediction by an AI system [90, 157, 158]. Nauta et al. [42] define confidence as the presence of a measure of certainty in an explanation. This makes the term confidence ambiguous since these contexts are all related to AI systems or explanations. Furthermore, some notions discussed in the the criteria of an explanation in Section 3.1.3 either share the same name or meaning as the concepts identified above. We have already mentioned that confidence is defined as the presence of a measure of certainty in the literature review from Nauta et al. [42], indicating that this term and/or notion appear in several other papers. Several other terms such as stability, robustness, continuity or sensitivity have been coined by the community to designate the idea that similar inputs have similar explanations [42, 74, 75, 90, 159]. The techniques that generate explanations are often described as functions that take the input, output and predictive function of an AI system as input and generates an explanation. Thus, the terminology attributed to mathematical function and used to characterize AI systems can be applied for explanations. Continuity is mostly intended for surrogate models, for which continuity implies that similar inputs lead to similar explanations [75, 159]. Likewise, sensitivity is employed for attribution-based explanation methods. It is a measure of the degree to which the explanation is affected by perturbations in the input of the AI algorithm to explain [102, 159]. Hence, continuity and sensitivity are ways to enforce stability or robustness in the case of attribution-based methods. Stability and robustness share the same meaning when qualifying explanations that is not specific to any kind of explanation technique. We observed that stability was preferred to robustness when addressing properties of explanations.

Finally, the term consistency often appears in the literature and is sometimes associated to stability. Indeed, Alvarez Melis and Jaakkola [75] define stability as a measure of how consistent the explanations are for similar/neighboring examples. Nauta et al. [42] state that "consistency evaluates whether identical inputs have identical explanations" which seems to be a special case of stability. They further clarify that consistency can address to what extent an explanation method is deterministic. Similarly, Carvalho et al. [90] explicitly distinguish stability from consistency. According to them, consistency compares explanations between different models. Particularly, it measures the difference between the explanations for two different models that have been trained on the same task and that output similar predictions. We noted in Chapter 3 that the term consistency had different meaning depending on the authors and the context. The Cambridge dictionary defines consistency as "the state or condition of always happening or behaving in the same way" which is similar to the definition of stability. Regarding the notion of consistency from Carvalho et al. [90] and Nauta et al. [42], the explanations for the two models should behave in the same way thus fitting the definition from the dictionary. Conversely, Förster et al. [43] consider an explanation consistent if it does not contain contain internal contradictions, hence using the definition of consistency in the field of logic i.e. consistency is the quality of containing no internal contra-

diction. This shows the ambiguity of the term consistency and demonstrates that further details about the type of consistency should be provided to clear up this ambiguity e.g. the internal consistency of an explanation may denote the definition of Förster et al. [43]. In any case, the term consistency has different meanings in fields related to XAI e.g. logic, knowledge bases, statistics and mathematics. Therefore, its definition should always be explicitly provided when employing this term.

Overall, we notice a convergence of the notions important for the XAI field. The current challenge of the terminology is caused by the sudden interest in XAI from a variety of research communities that all have a different vocabulary to denote the same general ideas. Notably, the machine learning community is particularly involved in the development of XAI; a domain that already has its own terminology that is itself inspired from computer science and mathematics. Due to the proximity between machine learning and XAI, the words employed to describe certain notions specific to XAI should be selected carefully to avoid ambiguities. The research articles that survey and review the XAI literature are the most exposed to the terminology issue and are generally required to make choices regarding the terms they identify for one concept. Unfortunately, considering the significant amount of papers produced in XAI, it is near impossible to propose a study of the terminology that is both exhaustive and didactic so that it is seen, understood and adopted by the community.

4.2 A terminology for a contextualized XAI

The literature review conducted in Chapter 3 and the previous sections provides a good overview of the several terms that occur regularly in the XAI field, along with their definitions and the ambiguities with other terms. Based on this review, we propose a terminology grounded in the context of explainable AI that maps one term to a single definition in an unambiguous way. The definitions are designed to be compatible with the majority of uses of the terms observed in the literature while also being close to the dictionary definition when possible to facilitate the comprehension and adoption of the terminology. The terminology is divided into two parts. The first part concerns the terminology related to qualities of an AI system in the context of XAI e.g. interpretability, robustness. The second part is the terminology of an explanation. It defines what an explanation is in the context of XAI and its different components as discussed in Section 3.1.1. We design an ontology for the system terminology, aligned with the DOLCE+DnS Ultralite (DUL) ontology [64]. Then, we instantiate the ontology design pattern to define explanations introduced in [65] to illustrate the explanation terminology.

In the introduction, we defined an AI system as a function that applies a particular decision process or AI algorithm to map an input to an output. This decision process is designed to perform a specific task. It can be viewed as a mathematical function that completes a task (e.g. image classification of animals, weather forecasting) based on its inputs. Thus, let f be a system such that $y = f(x)$ where y is the outcome or decision of the system and x is a set of input. The mathematical function itself corresponds to an AI algorithm ready to perform the task e.g. a trained neural network or a set of rules specific to the task.

Regarding the terminology for an AI system, we decided to use the same terms and definitions for both an AI system and an explanation technique. Indeed, an explanation technique has the same global functioning as an AI system, it takes a set of inputs and maps it to an output which corresponds to an explanation. The set of inputs contains at least the AI system to explain and can be enriched with the inputs used for the prediction in the case of local explanations. Other inputs can also be added such background knowledge depending on the explainability technique. Therefore, an explainability method can be regarded as a mathematical function similar to an AI system i.e. $e = \epsilon(f, x)$ where ϵ is the explainability method, f is the AI system to explain, x is the input of the AI system and e is the output of the method. Treating an explainability method like an AI system gives the opportunity to explain this method, which may be asked by a user to understand an explanation.

4.2.1 System terminology

The system terminology defines the properties of an AI system or an explainability method in the context of XAI and Responsible AI. Explainability and interpretability are the most important notions in the field of XAI and are employed to qualify an AI system. We propose definitions for these two terms and the related terms such as complexity or opacity. Then, we discuss the meaning of concepts related to Responsible AI e.g. transparency, reliability or robustness. Afterwards, the name and definitions of notions specific to explainability methods are identified. Finally, an ontology of an AI system and its qualities is introduced that summarizes this terminology.

Explainability and interpretability

The definition of explainability is straightforward and relies on the definition of an explanation that is provided later.

Definition 4.2.1 (Explainability)**.** Explainability is the ability of a system to explain itself or to be explained.

This definition of explainability stems directly from the construction of the word and relies on the definition of an explanation. We added the notion that an explainable system can either explain itself or be explained by another party, such as an explanation technique or a person. The proposed definition is very broad and it could be argued that any AI system is explainable thanks to post hoc explanations. Yet, we will see in the definition of an explanation that the explanations should satisfy the user needs. Moreover, we have seen that an explanation is an interactive process, meaning that an explainable system is able to interact with the user to explain. Hence, a system is truly explainable if it can interact with its users and adequately answer their questions which necessitates a variety of quality explanations.

Interpretability is mostly seen in the literature as an intrinsic quality of an AI system. Decision trees and linear models are generally considered naturally interpretable algorithms while neural networks are never seen this way. The review of the literature showed that interpretability is not intrinsic to the AI system [63] but rather depends on several factors that have not all been identified. Nevertheless, there is a link between the architecture of the AI algorithm and its interpretability, meaning that interpretability is a property of a system. The main difference between interpretability and explainability is that explainability implies an interaction with the user whereas interpretability is a passive property of a system. Likewise, based on our definition of explainability, an AI system can become explainable later in its life-cycle with the arrival of new explainability techniques. On the other hand, interpretability is static and never changes during the life of the AI system.

Scholars agree that the interpretability of a system depends on the user. The idea of cognitive effort is often used in the studied definitions of interpretability. It is also referred to as the limitations or the cognition, knowledge and biases of the user. Therefore, a system can be interpretable for one person and not for another. Based on these observations of how interpretability is defined and perceived in the literature, we propose the following definition.

Definition 4.2.2 (Interpretability)**.** Interpretability is the ability to be seen, studied and understood by a user with a reasonable cognitive effort.

An interpretable system is a system that can be studied by a user in order to understand its functioning and the resulting predictions. In other words, the user generates the explanations for themselves i.e. any question the user may have can be answered by studying the system. Thus, we argue that an interpretable system is explainable, since the system can be explained by another party that is the user. Our definition of interpretability can be applied to qualify an AI system but also any part of the system e.g. the input or output.

It was observed in the review that interpretability is strongly connected to the notions of transparency, simulatability, complexity and decomposability. The proposed definition of interpretabil-

ity keeps these connections. The ultimate goal of interpretability is that the user can fully understand the system without exterior intervention. To understand a system, the user has to study it and to do so, the mechanisms of the system should be available. The availability of the necessary information regarding the system is related to the transparency of the system. Furthermore, the notions of complexity and simulatability ensure that a reasonable cognitive effort is required to study and understand the system. Likewise, the decomposability and algorithmic transparency of the system guarantees that each part of the system is interpretable including its training phase, a necessary condition to enable the user to understand the system as a whole.

Definition 4.2.3 (Transparency)**.** A system is transparent if it provides all the information about its design and functioning for scrutiny.

We first define transparency as different concept from interpretability. We adopt the definition proposed by Beaudouin et al. [146] which is echoes the principle of transparency for Responsible AI identified by Mikalef et al. [27]. Transparency does not guarantee that a user is able to understand the system, but that they have access to all the information concerning the system e.g. the training data, how the data was processed, the performance of the system etc. Transparent systems are comparable to open-source software in which the code base can be seen and studied by anyone. This prevents the designer to use this program maliciously and is likely to increase trust in the system since experienced users can verify the quality of the design and detect any flaws or biases.

Definition 4.2.4 (Complexity)**.** Complexity is a measure of the interpretability of a system relative to its size.

Complexity is used in the literature as a measure of the size of a system which gives an indication on its interpretability. This measure is dependent on the AI algorithm exploited in the system, several measures can be proposed for the same algorithm. Several complexity measures have been identified in the literature as noted in Section 4.1.1. A large system requires a big cognitive effort which hinders its interpretability. Complexity is directly related to simulatability.

Definition 4.2.5 (Simulatability)**.** Simulatability is the ability of a system to be simulated or replicated by a user in a reasonable time.

Simulatability reflects the time required for a user to step through each calculation carried out by the system to make the prediction. The architecture of AI algorithms usually consist in repeating a same calculation e.g. linear models are a repetition of additions and multiplications, decision trees are a repetition of tests and neural networks are a repetition of matrix operations. Therefore, the simulatability depends on the difficulty to manually compute one iteration and the number of iterations to compute. Complexity impacts either or both of these factors depending on how the complexity measure is defined.

The notion of reasonable time in the definition of simulatability is similar to the reasonable cognitive effort in the definition of interpretability. The value of "reasonable time" is dependent on the user, their needs and backgrounds. Some users may desire to spend more time than others to understand a system which would render a system simulatable for these users while not simulatable for others. The time required to simulate the system's calculations is also dependent of the user's background knowledge and experience. Hence, a reasonable time implies a reasonable cognitive effort required for a system to be interpretable further connecting interpretability and simulatability. Yet, simulatability is not the only criterion necessary to ensure that a system is interpretable. Indeed, it is possible that user is able to simulate the system in a reasonable time without understanding what is being manipulated. Decomposability addresses this problem.

Definition 4.2.6 (Decomposability)**.** A system is decomposable if each of its components (e.g. inputs, parameters, calculations) are interpretable.

The main disagreement in the definition of decomposability was whether the components should be explainable or interpretable. Based on our observations in the literature review, decomposability appeared to be a criterion of the interpretability of a system. An interpretable system should not necessitate explanation, hence the components of the system should not require explanations either. Consequently, the components of the system should interpretable for the system to be decomposable.

The terms defined in this section are all related to explainability and interpretability. We noted that an interpretable system is explainable because interpretability implies that the user can explain the system on their own which fits the definition of an explainable system. Then, we proposed definitions for transparency, complexity, simulatability and decomposability that are all related to interpretability. Transparency guarantees that the system can be seen and studied, decomposability gives the assurance that every component of the system can be understood by the user. Simulatability ensures that the functioning of the system can be replicated in a reasonable time which enables the user to study and understand how the inputs are mapped to the output. Complexity is an objective measure of the size of the system which directly impacts the simulatability of a system and thus also impacts the interpretability of the system. The necessary conditions for a system to be interpretable are that the system is transparent, simulatable and decomposable. The interpretability of the components do not necessitate them to be simulatable since there is usually no calculations involved. Hence, the identified necessary conditions of interpretability only applies for a system and not its components.

Definition 4.2.7 (Opacity, black-box-ness)**.** An opaque or black-box system is a system that is not interpretable.

The definition of a black-box or opaque system was ambiguous because of the ambiguity between transparency and interpretability. We define a black-box system as the opposite of an interpretable system. This definition implies that the opacity of a system also depends on the user as it qualifies systems that are not interpretable which is dependent on the user. We also highlight that a black-box system may be explainable as is illustrated with the current efforts to "open the black-box" i.e. find methods to explain the behavior of a black-box system without modifying the system itself.

Finally, we did not define algorithmic transparency as we argue that it is included in several other definitions. Algorithmic transparency is the fact that the learning process of a system is visible and understandable. These conditions are already present in the definitions of transparency and decomposability. A system that is transparent and decomposable provides information regarding its learning algorithm and this algorithm is interpretable if the system is decomposable. Other definitions of algorithmic transparency were about the stability and reliability of the learning algorithm, which are notions that are defined in the following section about Responsible AI terms.

Responsible AI terms

We have seen in the literature review that some concepts related to the design of a Responsible AI reoccur in the literature. The lack of consensus for these concepts does not concern their definitions but rather their name. We first identify and define three concepts that relate to an AI system that are reliability, stability and robustness. Then, we discuss the definition and name of several other concepts that are specific to explainability methods e.g. consistency, faithfulness.

The definitions of stability and robustness to qualify a system are the definitions used in the related fields of machine learning. These notions participate in creating reliable systems that behave in an expected manner. The definition of stability or algorithmic stability is extracted from [29] and the definition of robustness is extracted from [155]. Both these notions refer to the ability of a system to behave in the same way when the inputs are slightly altered, either during training (stability) or in production (robustness). A notion that encompasses stability and robustness was identified in the literature to depict the assurance that a system is providing the correct answer

[77]. The terms confidence and reliability are usually associated to this notion. We selected the term reliability because we observed in the review that confidence is associated to several other meanings in XAI. Moreover, reliability is described as the generalization of robustness and stability by several scholars [28, 45, 67].

Definition 4.2.8 (Algorithmic stability)**.** A stable learning algorithm is an algorithm for which the learned solution does not change much with small changes in the training data.

Definition 4.2.9 (Robustness)**.** The degree to which a system or component can function correctly in the presence of invalid inputs.

Definition 4.2.10 (Reliability)**.** Reliability is the assurance that a system provides the correct answer and behaves in an expected manner.

The definition of reliability is inspired from the dictionary definition and the one of Calegari et al. [77]. The Cambridge dictionary defines reliability as "the quality of being able to be trusted or believed because of working or behaving well". Calegari et al. [77] define a reliable system as a system that provides the correct answer. We distinguish two requirements for reliability: 1) the system generally provides the correct answer in ideal conditions and 2) the system behaves well in any circumstance which implies that it always performs as it performs in ideal conditions. A system that behaves in an expected manner but never provides the correct answer is not reliable. Consequently, stability and robustness are directly related to reliability as they give the assurance that the system behaves in an expected manner even when the inputs are altered or unexpected. As a result, we consider that stability and robustness of a system are necessary conditions to consider a system reliable. The fact that the system provides the correct answer depends on its performance which is expected to be satisfactory if the system is moved to production.

Next, we define terms specific to explainability methods that relate to the quality of the explanations generated. We discussed in the beginning of this section that explainability methods can be viewed as a system that takes an AI system as input and generates an explanation as the output. Therefore, the AI system terminology applies to explainability methods. Nevertheless, some notions are specific to explainability methods. Although there exists a large number of vocabulary to qualify an explainability method, we limit our terminology to the terms that are ambiguous with regards to the terms defined in the terminology and those employed in this thesis.

We first address the concept that depicts how accurate the explanation is with regards to the system to explain. Several terms are associated to this notion: soundness, truthfulness, correctness, fidelity and faithfulness. We prefer the terms faithfulness and fidelity which are synonyms in the dictionary. The term soundness has many meanings in the English dictionary, one of them is "the quality of being able to be trusted" which fits the notion to describe. Still, other meanings of soundness relate to the completeness of something which may create ambiguities. The terms truthfulness and correctness can be used to refer to the correctness of a prediction from an AI system which can also lead to ambiguities. Finally, faithfulness and fidelity are synonyms that both carry the idea of being true to something. These terms are not used in XAI or related fields, therefore they are ideal candidates for the discussed concepts. We use the definition of fidelity proposed by Yang et al. [40].

Definition 4.2.11 (Faithfulness, fidelity)**.** The ability of an explanation to precisely capture the decision making process and show the correct evidences

Similar to AI systems, scholars are interested in creating explanations that are not affected by small changes in the inputs of the explainability method that do not affect the final prediction. The inputs of these explainability methods are the system to explain and the input of the system to generate a prediction. Stability in the context of explainability methods guarantees that the explanations remain similar when the input of the system to explain are slightly different. Sensitivity is another term that is used with the same meaning as stability for explainability methods. In addition, the stability or sensitivity of an explanation is not always desired contrary to AI system where

stability is an ideal characteristic. Therefore, we use the term sensitivity to avoid ambiguity with stability defined earlier. The notion of continuity is related to sensitivity as it is a mathematical property of an explainability method that participates in reducing sensitivity. We do not propose a definition for continuity as it corresponds to the standard mathematical definition of a continuous function.

Definition 4.2.12 (Sensitivity)**.** Sensitivity is the measure of how much the explanation changes when the input of the system to explain changes.

While sensitivity measures how the explanation is affected by changes in the input of the system, another term must be determined to refer to the measure of how the explanation is affected by changes in the system while retaining the same prediction. Some scholars define consistency as a comparison of explanations between different systems that realize the same task with the same data [42, 90]. We observed that the term consistency is an ambiguous term as it is used in many other domains related to XAI. Moreover, we employ the term consistency to refer to the consistency of an ontology i.e. whether an ontology contains any contradiction. The notion of implementation invariance is cited in the literature as the fact that two models that give the same outputs for all inputs should have the same explanations [90]. Hence, we select this term to name this notion. We note that the sensitivity and implementation invariance concepts are both defined in [90] specifically for neural networks and attribution-based explanation methods. We propose a generalize variant of these definitions.

Definition 4.2.13 (Implementation invariance)**.** An explainability method is implementation invariant if for two equal systems (i.e. systems that complete the same task and return identical predictions), the generated explanations are identical.

In the proposed system terminology, we have identified and defined notions that qualify an AI system and/or explainability methods. This terminology is not exhaustive but rather focuses on the terms and concepts that were ambiguous because of their naming or their definition in the literature. In addition to the terminology, we established connections between the different studied notions e.g. an interpretable system is necessarily transparent, decomposable and simulatable. In the following section, we design an ontology aligned with the DOLCE+DnS Ultralite ontology [64] that represents these notions and their connections.

AI system ontology

The proposed AI system terminology revealed some connections among notions related to an AI system. We materialize these connections by designing an ontology[3] of an AI system aligned with the DUL ontology [64]. We ignored explainability methods and their specific notions in the ontology to focus only on the general concept of an AI system. The class `AISystem` is defined as a subclass of `dul:Agent`. The AI system can have specific qualities represented by instances of the class `AISystemQuality` which is a subclass of `dul:Quality`[4]. These qualities represent every notion discussed in the terminology e.g. explainable, decomposable, robust. They are represented as different instances of the class `AISystemQuality` which is itself a subclass of `dul:Quality`. 8 qualities of an AI system are defined: decomposable, explainable, interpretable, reliable, robust, simulatable, stable and transparent. The classes `AISystem` and `AISystemQuality` and the different qualities are defined as follows:

$$\texttt{AISystem} \sqsubseteq \texttt{dul:Agent} \tag{4.1}$$

$$\texttt{AISystemQuality} \sqsubseteq \texttt{dul:Quality} \tag{4.2}$$

[3] https://git.litislab.fr/s4xai/xai-terminology-ontologies/-/blob/main/AISystem.owl

[4] `dul:Quality` is defined as any aspect of an `Entity` (but not a part of it), which cannot exist without that `Entity`.

$$\texttt{AISystemQuality(decomposable)} \tag{4.3}$$
$$\texttt{AISystemQuality(explainable)} \tag{4.4}$$
$$\texttt{AISystemQuality(interpretable)} \tag{4.5}$$
$$\texttt{AISystemQuality(reliable)} \tag{4.6}$$
$$\texttt{AISystemQuality(robust)} \tag{4.7}$$
$$\texttt{AISystemQuality(simulatable)} \tag{4.8}$$
$$\texttt{AISystemQuality(stable)} \tag{4.9}$$
$$\texttt{AISystemQuality(transparent)} \tag{4.10}$$

Based on these two main classes and eight qualities, we define the classes that represent the different AI systems that have these qualities:

$$\texttt{DecomposableAISystem} \equiv \texttt{AISystem} \sqcap \exists \texttt{dul:has\_quality.}\{\texttt{decomposable}\} \tag{4.11}$$
$$\texttt{ExplainableAISystem} \equiv \texttt{AISystem} \sqcap \exists \texttt{dul:has\_quality.}\{\texttt{explainable}\} \tag{4.12}$$
$$\texttt{RobustAISystem} \equiv \texttt{AISystem} \sqcap \exists \texttt{dul:has\_quality.}\{\texttt{robust}\} \tag{4.13}$$
$$\texttt{SimulatableAISystem} \equiv \texttt{AISystem} \sqcap \exists \texttt{dul:has\_quality.}\{\texttt{simulatable}\} \tag{4.14}$$
$$\texttt{StableAISystem} \equiv \texttt{AISystem} \sqcap \exists \texttt{dul:has\_quality.}\{\texttt{stable}\} \tag{4.15}$$
$$\texttt{TransparentAISystem} \equiv \texttt{AISystem} \sqcap \exists \texttt{dul:has\_quality.}\{\texttt{transparent}\} \tag{4.16}$$
$$\texttt{InterpretableAISystem} \equiv \texttt{AISystem} \sqcap \exists \texttt{dul:has\_quality.}\{\texttt{interpretable}\} \tag{4.17}$$
$$\texttt{ReliableAISystem} \equiv \texttt{AISystem} \sqcap \exists \texttt{dul:has\_quality.}\{\texttt{reliable}\} \tag{4.18}$$

Two classes have additional definitions, reliable and interpretable AI systems. Indeed, it was stated that an interpretable AI system is necessarily decomposable, simulatable and transparent. Furthermore, an AI system that is interpretable is necessarily explainable. Similarly, it was stated that a reliable system is necessarily robust and stable. These additional descriptions are given in the following definitions:

$$\texttt{InterpretableAISystem} \sqsubseteq (\texttt{DecomposableAISystem} \sqcap \texttt{SimulatableAISystem} \sqcap \texttt{TransparentAISystem}) \sqcup \texttt{ExplainableAISystem} \tag{4.19}$$
$$\texttt{ReliableAISystem} \sqsubseteq \texttt{StableAISystem} \sqcap \texttt{RobustAISystem} \tag{4.20}$$

Finally, the class `BlackBoxAISystem` is defined as any AI system that is not interpretable. We note that this definition allows a black box AI system to be decomposable, transparent, explainable and/or simulatable. Its corresponding ontology definition is:

$$\texttt{BlackBoxAISystem} \equiv \texttt{AISystem} \sqcap \neg \texttt{InterpretableAISystem} \tag{4.21}$$

This ontology enables the XAI community to characterize an AI system, by attributing it several qualities. An AI system that is interpretable is automatically transparent, decomposable and simulatable according to the ontology. Likewise, a reliable system is necessarily stable and robust. Nonetheless, the sufficient conditions to create an interpretable or reliable AI system are not clear which explains why we use a subclass relation instead of a equivalence relation between the types of AI system.

4.2.2 Explanation terminology

The explanation terminology explores terms that are specific to the design of explanations in the context of XAI. The definition of an explanation has already been largely discussed in Section 3.1.1 but we did not settle on specific definitions that are employed in the rest of this thesis. The terminology related to explanations is less controversial because explanations are not investigated by

the communities related to XAI such as the machine learning community. Thus, most terms are either new and unambiguous or were already extensively researched by other fields interested in explanations e.g. psychology or sociology. In this section, we explicitly define an explanation and discuss related terms such as post hoc/ante hoc explanations. Then, we instantiate and extend the ontology design pattern for explanations from Tiddi et al. [65] to depict an explanation in the context of XAI.

Explanation terms

Definition 4.2.14 (Explanation)**.** An explanation is the result of an interaction between an explainee and an explainer, during which the explainer provides relevant causes of a phenomenon that are understandable by the explainee.

In Section 3.1.1, we discussed the proposed definitions of an explanation and identified the main components and actors of an explanation. Definition 4.2.14 is the definition that emerged from this discussion. The two actors of an explanation are the explainee and explainer and have been defined above. The technical terms for the components have not been integrated to the definition because they are not commonly used. The causes provided by the explainer are the *explanans* and the phenomenon to explain is the *explanandum*. As illustrated in Figure 3.1, the explanans and explanandum are linked together with some causal reasoning e.g. the floor is wet (explanandum) because it is raining (explanans) and it is known that rain causes things to be wet (causal reasoning). In this example, the last part that links the explanans to the explanandum is not necessary because it is common knowledge. However, it may be necessary to explicitly provide the reasoning that connects the explanans to the explanandum in cases where the explainee does not have the required knowledge to implicitly understand it.

We now focus on explanations in the context of XAI. The goal of an explanation in XAI is to determine relevant causes to explain an AI system and its outcome that are understandable by a user. There are several possibilities regarding the nature of the explanandum, the method to determine the explanans and the causal reasoning to link the explanans to the explanandum. The nature of the explanandum refers to the part or action of the AI system to explain. In XAI, two types of explanandum are generally addressed: the global functioning of the AI system or a particular outcome of the AI system. The former explanandum calls for a global explanation while the latter requires a local explanation. Global and local explanations are two notions that are well defined in the literature.

Definition 4.2.15 (Global explanation)**.** A global explanation describes the functioning of the entire AI system.

Definition 4.2.16 (Local explanation)**.** A local explanation identifies the causes that led to a specific outcome of an AI system.

The explanans of an explanation in the context of XAI can be obtained in several ways. The role of explainability methods is to determine an adequate explanans. When the AI system has qualities such as simulatability, decomposability or even interpretability, it is possible to directly extract the explanans from the system. The explanations generated by directly extracting relevant causes from such AI systems are called *ante hoc* explanations. Other explainability methods are applied to find explanans that do not directly exploit the architecture of the AI system. The explanations designed without exploiting the AI system functioning are called *post hoc* explanations.

Definition 4.2.17 (Ante hoc explanation)**.** An ante hoc explanation is an explanation that directly exploits the AI system mechanism to determine the explanans.

Definition 4.2.18 (Post hoc explanation)**.** A post hoc explanation is an explanation that uses an auxiliary method to determine the explanans.

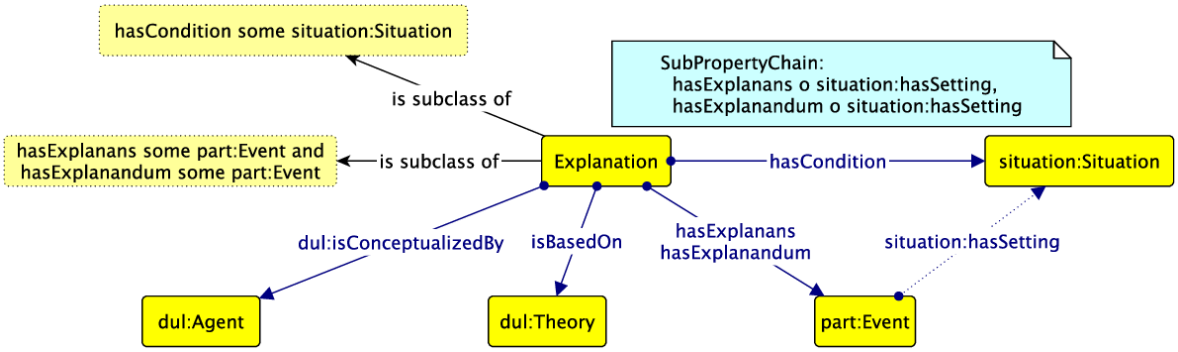


Figure 4.2: The ontology design pattern to define explanations [65].

Other notions linked to explanations in the context of XAI have been identified in the literature, such as model-specific or model-agnostic explainability methods as well as the type of reasoning that connects the explanans to the explanandum. These notions have already been discussed and defined in Chapter 3 as their definitions and names make consensus. In this same chapter, we mentioned the ontology design pattern to define explanations proposed by Tiddi et al. [65]. In the following section, we instantiate this design pattern to represent explanations specific to XAI and extend it to reflect the proposed terminology.

4.2.3 Formalization of XAI explanations

The ontology design pattern (ODP) to define explanations was introduced by Tiddi et al. [65] and is displayed in Figure 4.2. It is a generalization of the definitions of an explanation proposed in different domains. They define an explanation as an entity that possesses at least an antecedent event and a posterior event that both happen in the same context. The antecedent event is the explanans and the posterior event is the explanandum that are both instances of the class `Event` from the Participation ODP[5]. The context in which both the explanans and explanandum take place is represented as an instance of the class `Situation` from the Situation ODP[6]. Two remaining components of an explanation are the agent that conceptualizes the explanation and the theory that binds the explanans to the explanandum, both depicted with the classes `Agent` and `Theory` from the DUL ontology [64]. We note that the Situation and Participation ODP are both directly extracted from DUL. In the same paper, they instantiate this ODP to describe explanations in the context of different research fields. Thus, we propose to instantiate this ODP for explanations in the XAI context.

We have discussed the components of an explanation in XAI and the possible choices for each component. The explanandum is related to the behavior of an AI system i.e. its global functioning or a particular prediction. This AI system is designed to perform a specific task i.e. it has devised an algorithm or plan to produce the desired outcome based on a set of inputs. In the AI system ontology, an AI system is a subclass of `dul:Agent` and in the DUL ontology [64], a situation is the result of the execution of a plan by an agent. Therefore the situation in which the explanation takes place is the execution of the task by the AI system. The explanans is the output of an explainability method applied to the AI system that created the explanandum, hence the explanans and explanandum share the same situation. The agent that conceptualizes the explanation is not determined, since we did not specify the agent responsible of generating the explanation in the definition of explainability. Still, we highlighted that the explanation was generated by the explainee when the system is interpretable. In most cases, it is implicit that the explanation should be created by the AI system itself to avoid the need of human intervention to explain every prediction. Thus, the agent that conceptualizes the explanation is either the explainee or the AI system. Finally, the theory exploited to causally connect the explanans to the explanandum is described in

[5] `http://ontologydesignpatterns.org/cp/owl/participation.owl`
[6] `http://www.ontologydesignpatterns.org/cp/owl/situation.owl`

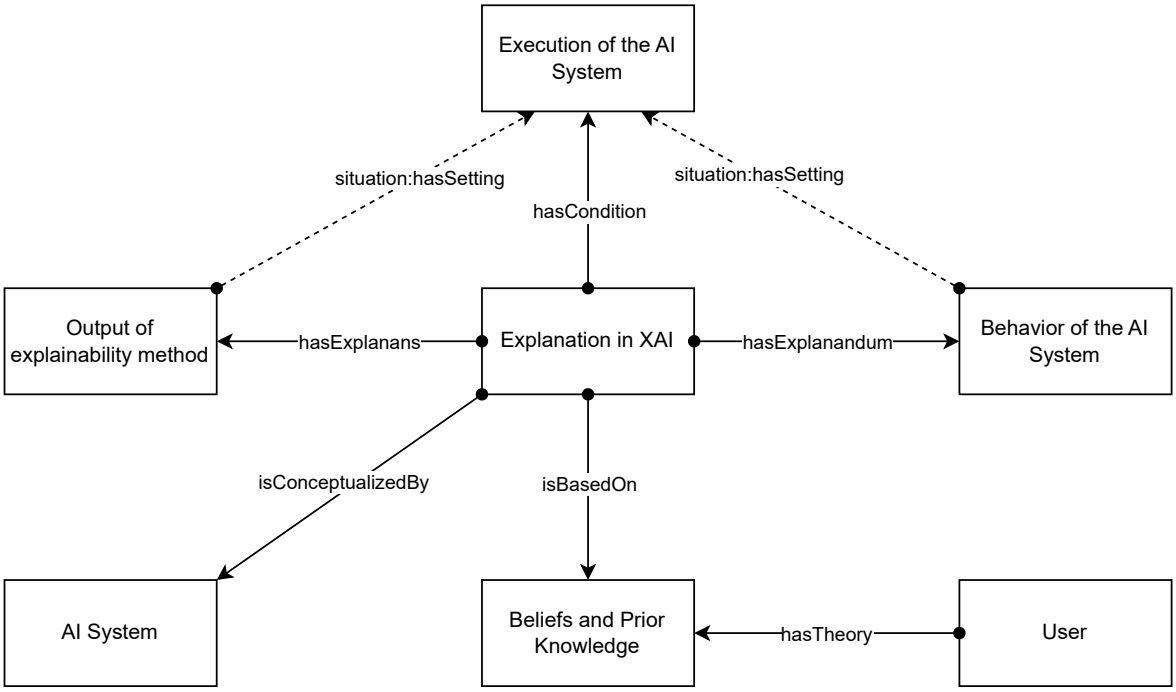


Figure 4.3: ODP to define explanation for explanations in XAI

the documentation as "a set of assumptions for describing something, usually general. Scientific, philosophical, and commonsense theories can be included here". In the context of XAI, we mentioned that the explanation should be adapted to beliefs and prior knowledge of the explainee. Consequently, the theory in the context of XAI is the beliefs and prior knowledge of the explainee. This has the side-effect of including the explainee in this instance of the pattern which addresses our main problem with this ODP (see Chapter 3). The causal reasoning that was discussed in the definition of an explanation is already contained in the theory component.

The application of the ODP is shown in Figure 4.3. This figure follows the schematic given in [65] to instantiate the explanation ODP. The class of AI system is the class `AISystem` as defined in the system ontology, which fits with the ODP since it is a subclass of `dul:Agent`. The AI system has an influence on the situation, the explanans and explanandum though this influence is not illustrated in this instance. The several types of explanations (e.g. global or local, ante hoc or post hoc) are special cases of this pattern where the explanans or the explanandum are refined. For instance, a local explanation follows the same pattern but the explanandum is specifically a single prediction of the AI system.

4.3 Conclusion

In this chapter, we studied the vocabulary employed in the literature to identify the recurrent concepts and terms of XAI and ambiguities regarding the name or definition of these concepts. We noticed that most ambiguities were caused by the fact that XAI attracts different communities that do not share the same terminology to describe identical concepts. Particularly, the terminology of machine learning is widely used but rarely defined in the context of XAI. As a result, the terminology to characterize XAI concepts for an AI system is ambiguous, with the application of several terms associated to a same notion, or several notions associated to a single term.

We address this problem by proposing a terminology based on our observations of the literature. The terminology is divided into two parts, one regarding the terms of an AI system and the other regarding the definition of an explanation. Regarding the AI system terminology, we defined commonly used terms such as interpretability and explainability and identified relations between these terms. Explainability methods are considered as particular AI systems that share the same

characteristics while having their own set of specific properties such as faithfulness or sensitivity. This terminology highlighted particular relationships between notions related to the interpretability of a system. These relations are formalized in an ontology of an AI system that is aligned with the DUL ontology [64]. This ontology is designed to facilitate the comprehension and adoption of this terminology.

The second part of the terminology concerns the definition of an explanation. It is an extension of the discussion in Section 3.1.1 where we propose our own definition of explanation. We particularly emphasize that an explanation is the result of an interaction between the explainee and the explainer. We further discuss the components of an explanation (i.e. explanans and explanandum) in the context of XAI and briefly define global/local and post hoc/ante hoc explanations. Most notions related to the characteristics of an explanation are already well defined in the literature and were not explored in this terminology. Finally, we positioned our definition of an explanation with regards to the ontology design pattern to define explanations by Tiddi et al. [65]. We instantiated this design pattern to represent explanations in XAI with the help of the ontology of an AI system introduced in the first part of the terminology. Notably, we describe the explanans (the causes that provoked the event to explain) as the output of an explainability method and determined the influence of the user on an explanation. Indeed, scholars agree on the fact that the explanation depends on the user yet this dependence is not explicit in the ODP to define explanations. The theory or set of rules that is exploited to connect the explanans to the explanandum is depicted as the beliefs and prior knowledge of the user. Hence, the explanation is based on this theory that directly depends on the user. Still, this pattern does not reflect the interactive nature of an explanation but rather shows the explanation at the end of this process.

Overall, the proposed terminology uses broad definitions that enable their application on any type of AI system and do not contradict the definitions introduced in other papers. It focuses on recurrent terms specific to XAI that presented ambiguities in some way, either because of their name or their definition. Still, the proposed terminology presents some drawbacks that are inherent to this exercise. Firstly, it is not exhaustive as it is nearly impossible to identify every notion employed in the XAI domain and map these notions to adequate names and definitions. Secondly, the terminology and ontology only reflects our comprehension and view of these terms in the context of XAI. A good terminology is a terminology that is understood and adopted by the entire community. Hence, this contribution is only a proposition that is open to debate while avoiding ambiguities in the rest of this thesis by explicitly defining the vocabulary. Finally, we observed a quick evolution of the terminology in the 2-year time frame that separates our article regarding a terminology (published in 2021) and the writing of this thesis. The terminology of XAI seems to converge towards a unique shared terminology. In the meantime, the work that aims at identifying and measuring the components of interpretability and explainability are making progress and thus introduce new notions to the XAI terminology. These notions may replace or refine older notions, including some defined in this terminology. Therefore, our proposition of a terminology is destined to become obsolete in the next years; indicating that the XAI terminology is evolving towards a consensual terminology similar to mature scientific fields.

Chapter 5

An ontology-based explainable intelligent system to classify images

Contents

5.1 Literature review . . . 76
5.1.1 Error detection . . . 77
5.1.2 Explainable image classifiers . . . 78
5.1.3 Explainable neurosymbolic methods . . . 82
5.1.4 Explanation interfaces . . . 83
5.1.5 Discussion . . . 84
5.2 OBIC: explainable ontology-based image classifier . . . 85
5.2.1 Ontology requirements . . . 87
5.2.2 Training phase . . . 87
5.2.3 Inference phase . . . 90
5.3 Explanations with OBIC . . . 92
5.3.1 Extraction of the explanations . . . 92
5.3.2 Design of the explanation interface . . . 94
5.4 Conclusion . . . 96

An architecture for an XIS was proposed by DARPA [15] and is shown in Figure 5.1. Similar to a standard machine learning model, it requires training data and a process to train the model. However, the resulting model enables the generation of explanations about its output or functioning. Since an explanation is an interactive process, an explanation interface is added to the AI system to adequately answer the user's questions.

Current XAI methods focus on the first part of the system i.e. creating explainable models with new machine learning process. Likewise, designs for explanation interfaces have been proposed meaning that explainable intelligent systems can be assembled by combining these separate works. Yet, explanation interfaces rarely use explainable models to generate explanations but rather use well-known post hoc methods to explain a black box model. To our knowledge, there is little to no explanation interfaces that extract explanations based on explainable or interpretable models meaning that the design of a complete XIS has not been studied yet.

In Chapter 2, we discussed the several goals of XAI. We observed that current XAI techniques focus on fairness and transparency at the expense of XAI methods for accountability and safety. Adadi and Berrada [29] discussed about explaining to control i.e. explain to detect and prevent errors and system failures. Most explainable models use machine learning in some way to make a prediction or generate an explanation. These methods are built on the assumption that the underlying machine learning model will give the correct prediction. Consequently, the cases where

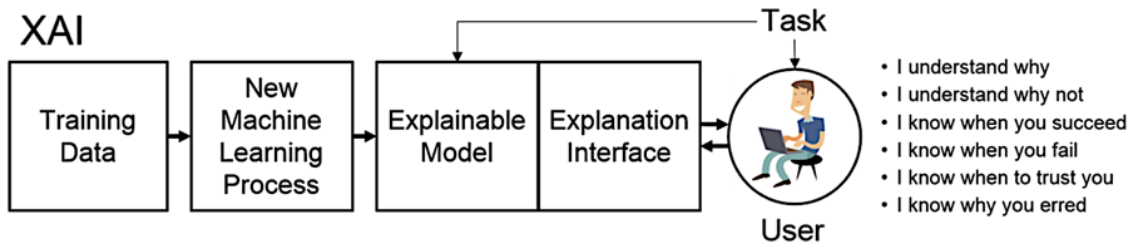


Figure 5.1: An explainable AI system as described by DARPA [15]

the model is wrong are ignored. This may cause misleading or unfaithful predictions as well as explanations that lead to undesirable consequences.

Finally, it was discussed in Section 3.1.2 that neurosymbolic approaches are ideal candidates to create interpretable models with state-of-the-art performance. Particularly, the integration of ontologies to current machine learning algorithms is being actively researched. We have also mentioned that the current work on XAI that use ontologies are especially focused on integrating ontologies to neural networks. Model-agnostic XAI solutions based on ontologies are currently lacking. These observations motivated us to create a complete XIS that is model-agnostic and based on ontologies. We designed this XIS for the task of image classification as the majority of current XAI techniques can be applied to this task. Hence, it enables the comparison between our contribution and the state of the art XAI methods.

In this chapter, in Section 5.1 we review the literature on error detection methods, explainable image classifiers, explainable neurosymbolic models and explanation interfaces. Then, in Section 5.2, we introduce our design of an XIS on the task of image classification, capable of detecting and explaining errors in its predictions. An explainable model is created by using knowledge from an ontology to automatically build and train machine learning models. Then, the predictions from these models are combined into an individual that is added to the same ontology. A logical reasoner is applied to verify the consistency of the predictions, acting as an error detection step. Finally, in Section 5.3, we discuss the explanations that can be extracted from this model and the consequent design of an explanation interface, with the goal of making the explanations comprehensible to any user.

5.1 Literature review

There exists a large variety of explainability techniques that have been compiled into toolkits [34, 73, 160, 161]. These toolkits are mostly Python libraries that are designed exclusively for AI practitioners. Bhatt et al. [162] held a day-long discussion with academics, industry experts, legal scholars and policymakers about explainability. The different parties mentioned the importance of being able to interact with explanations as well as providing the uncertainty of a prediction alongside an explanation. Similarly, Chazette et al. [163] conducted a literature study and interviewed 19 industry experts to get insights on how to develop explainable systems. The main takeaway from these studies is that the development of explainable systems should be user-centered i.e. the needs of the end-user should be included at every stage, especially when determining explainability requirements.

In the following, we study the literature on error detection to assess how uncertainty can be included in an XIS. Then, we review the state of the art on explainable image classifiers and neurosymbolic methods. Finally, we survey the literature on explanation interfaces and how they can adapt the explanations to the end-user's needs.

5.1.1 Error detection

Machine learning models are mathematical functions, thus they always make predictions even when they are likely to be inaccurate. Research on detecting errors and estimating the confidence of a prediction on unseen data is a response to this problem. Most approaches evaluate the confidence of a prediction and are able to predict a failure. The output of a classifier is commonly interpreted as the confidence that a class is present in the input. However, scholars observed that machine learning classifiers and especially neural networks often fail silently i.e. they provide high confidence predictions even when incorrect [157, 158, 164, 165]. Therefore, more reliable ways to compute the confidence or uncertainty of a prediction need to be developed.

The objective of error detection is to be able to accurately predict when a model will be wrong. Hendrycks and Gimpel [164] stated that an error detector classifies a prediction into two classes: positive (i.e. correct) and negative (i.e. incorrect). Evaluating an error detection system is therefore equivalent to evaluating a binary classifier. If the training data is imbalanced and contains far more negative classes than positive classes (e.g. the classifier mostly makes correct predictions and makes few errors), then the detector will always guess that there is no error and achieve a high accuracy. Hence, the study of false positives and false negatives is much more relevant than studying the accuracy. Consequently, a common metric in this domain is the AUROC metric which takes into account the false positive and false negative rates. In addition, they used the AUPR metric to handle some cases where the AUROC behaves poorly. In the same paper, Hendrycks and Gimpel [164] introduced a baseline method to measure the confidence of a classifier. This method, called Maximum Class Probability (MCP), takes the value of the predicted class's probability. As we discussed earlier, this method is not ideal and has conceptual drawbacks [158].

Jiang et al. [157] introduced the *trust score* to determine a classifier's trustworthiness for a particular input. High-density sets or clusters are created for each class based on the training data. Then, the *trust score* is calculated as the ratio between the distance from the test sample to the predicted class and the distance from the test sample to the high-density set of the nearest class that is not the predicted class. If the classifier is correct, the high-density set of the predicted class should be closer to the test sample than any other set. A high *trust score* implies a trustworthy prediction and therefore a correct prediction. Inversely, a low *trust score* means that the prediction is uncertain and probably wrong. This model-agnostic method consistently shows better results than the model confidence except in high-dimensional cases where the score provides little improvement over the model confidence.

Corbiere et al. [158] proposed an improvement of the MCP approach discussed earlier. For a given input x, the MCP estimates the probability of the predicted class which corresponds to the maximum probability in the output of the model. Using the maximum probability of the model systematically leads to high confidence values for both correct and incorrect predictions. Conversely, the probability of the true class is likely to be a low value in the case of incorrect predictions. Therefore, the true class probability (TCP) is a better choice to accurately reflect an error. However, the true class is generally not known meaning that the TCP has to be approximated. The authors proposed to estimate the TCP with a neural network, using the training data and the output of the classifier. Then, they applied thresholds on the TCP value to determine whether a prediction is correct or incorrect. Their method performs better than MCP and the *trust score*. Nevertheless, TCP and MCP are not model-agnostic but are designed for neural networks.

There exists a different approach to error detection which consists in creating a "check model" that has the same task as the model [165, 166]. An input is considered misclassified by the original model if the "check model" and original model disagree. Chen et al. [165] used an ensemble of models that each classify the same input. An input is misclassified if the majority of the models in the ensemble disagree with the original model. They evaluated their method using only the F1-score and compared against MCP and *trust score*. The method consistently achieves the best performance, yet it is not possible to compare against TCP [158] because of the different metrics used in the respective papers.

Zhang et al. [167] linked the field of uncertainty quantification to XAI. They noticed that cur-

rent explainability methods are not sufficient for decision-making in safety-critical environments. They added that end-users are interested in learning which factors contribute to the prediction uncertainty. As a result, they proposed a method for computer vision that analyzes the contribution of each pixel to prediction uncertainty and then uses this information to reduce uncertainty. The maximum entropy of a prediction is computed with a Bayesian neural network and compared to a threshold. If the entropy is greater than the threshold, the model says "I do not know" and shows the feature importance and uncertainty of each pixel with regard to the predicted class. Finally, a technique to reduce model prediction uncertainty by modifying pixels with a high uncertainty is proposed.

Error detection methods are able to prevent wrong predictions by giving a model the ability to say "I am not sure". These methods usually evaluate the confidence or uncertainty of a prediction and choose to reject it based on a fixed threshold. Zhang et al. [167] applied this strategy to increase explainability and safety of any model. They argued that users are interested in learning the contributing factors to uncertainty. Indeed, rejecting a possibly wrong prediction is a step towards safer and more accountable systems. We argue that understanding the reasons of such rejection is as important. The current solutions are not designed with such a goal, rendering difficult or even impossible to explain why a prediction was rejected. The method proposed by Zhang et al. [167] introduces a way to explain the uncertainty but pixel-wise explanations are considered poorly interpretable [75].

5.1.2 Explainable image classifiers

Computer vision models for image classification or segmentation are used in safety-critical fields such as medicine or autonomous cars. Although these fields require explanations for every decision, the state-of-the-art computer vision models are neural networks that notoriously lack explainability. Therefore, the design of XAI methods dedicated to computer vision is being actively studied. We restrict this literature review to the problem of generating local explanations of image classifiers.

The most common approach to explain image classifiers is feature attribution [47, 48]. The output of this type of explanation is a visualization of the importance of each pixel for a prediction with regard to the predicted class. This visualization is also often called *saliency maps*. Additive feature attribution methods (see Chapter 3) such as LIME [49], SHAP [50] or DeepLIFT [51] are model-agnostic and thus can be applied to image classifiers. Several methods are designed specifically to extract feature attributions for image classifiers. Two approaches are mainly used: occlusion-based and gradient-based methods [47]. Occlusion-based methods modify parts of the image to study the resulting difference in the prediction. Based on the difference in prediction score, these methods extract the feature attribution of the modified parts. Different strategies are elaborated to choose the zones of an image to perturb and how to perturb these zones.

- The RISE method [168] functions by randomly sampling and deleting pixels of the input image (i.e. setting them to a black pixel), then studying the impact on the prediction and finally determining the importance of each pixel based on this impact.
- Zhou et al. [169] iteratively perturbed zones of an image while preserving the class score. Instead of setting the pixels to black, the perturbed zones are set to their average color. After a number of iterations, the remaining zones of the image are the one that significantly impact the prediction.
- Dabkowski and Gal [170] proposed a different approach by training a model to determine the zones of importance instead of relying on a iterative process. The authors described this method as faster than the iterative methods while also producing higher quality saliency maps.

The advantage of these methods is that they are model-agnostic, contrary to gradient-based methods that are only applicable to neural networks.

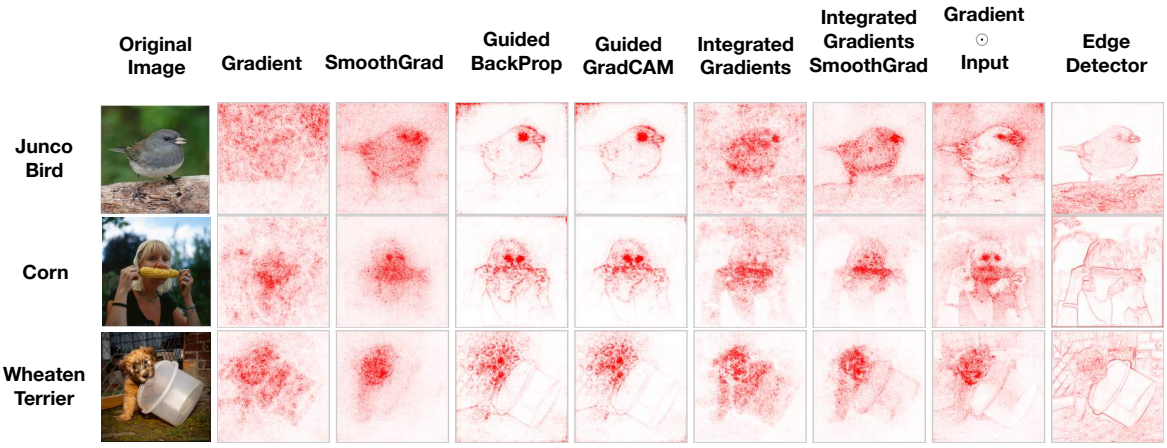


Figure 5.2: Saliency maps for some common methods compared to an edge detector [52].

Gradient-based methods compute the regions of importance of the image by exploiting the gradient of the predicted class. Simonyan et al. [53] introduced this class of method by computing the gradient through back-propagation. Important regions of the image are determined by the gradient's magnitude for each pixel. Similar approaches have followed, that built upon this idea ([171, 172]). *Class Activation Mapping* (CAM) [173] is another approach that extracts the regions of an input image used by a CNN to predict a given class. This method requires the classifier to be a CNN that does not contain any fully-connected layers. Grad-CAM [174] was later proposed as generalization of CAM and is applicable to a variety of CNN models including CNNs with fully-connected layers.

Different scholars studied the validity of the saliency maps approach. Adebayo et al. [52] presented an evaluation of these methods to test whether the saliency maps are sensitive to either data or model. Sensitivity to data or model is highly desirable for these methods as their goal is to show what a particular model looked at when classifying an image. They compared the output of these methods with a simple edge detector in Figure 5.2. They argued that the results of saliency methods are similar to the output of an edge detector. This observation led to a discussion about the risk of confirmation bias from the human observer when interpreting saliency maps. As Molnar [54] mentioned, it is difficult to know whether an explanation is correct. Adebayo et al. [52] noted that saliency maps method could implicitly implement image processing techniques which provides pleasant explanations at the expense of their faithfulness. Furthermore, Molnar [54] remarked that the current state of saliency maps is very unsatisfactory because of the proven fragility and unreliability of these methods combined with a lack of proper evaluation tools to assess their faithfulness.

In response to these issues, Nguyen et al. [175] introduced the ObAlEx metric to evaluate explanations of image classification models. To compute this metric, an image is given as input of an image classifier and an object detector. The object detector outputs a mask that outlines the regions where the object is detected. In parallel, the prediction of the image classifier is explained with a saliency map. Regions of the explanation that lie outside the object mask are considered indicative of a classification for the wrong reasons. ObAlEx corresponds to the sum of the importance of pixels inside the object mask divided by the sum of the importance of pixels in the entire image. A perfect ObAlEx score of 1 means that only pixels inside the object mask have a non-zero importance according to the saliency map. The explanation quality of an image classifier can be obtained by computing the ObAlEx score on all images in a dataset and calculating the average score for correctly classified images.

With similar motivations, Vermeire et al. [176] proposed a method to generate counterfactual explanations[1] for image classifiers. They observed that counterfactual explanations are better

[1] A literature review on counterfactual explanations is conducted in Section 6.1.

suited to explain an impactful decision to end users. We note that counterfactual explanations are closely related to the occlusion-based methods. Indeed, both explanations seek a set of modifications that lead to a different prediction. However, in the case of counterfactual explanations, this set of modifications constitutes the explanation. Hence, it must be comprehensible and avoid perturbations that are indistinguishable by human observers. The SEDC method proposed by Vermeire et al. [176] separates the image in a certain number of segments. Then, minimal combinations of segments that change the prediction when removed are found. Each combination of segments constitute a counterfactual explanation.

Counterfactual and contrastive explanations for an image classifier are provided in a textual format by the method of Pintelas et al. [55]. They used a completely interpretable pipeline to predict and explain. First, a feature extraction framework is designed to extract a set of interpretable features in an image that is then given to a linear model to train it. Instead of using black-box models such as CNNs for feature extraction, they manually chose a set of features that are understandable by humans and useful for the linear model. Image processing techniques are then employed to extract texture and contour features. This set of features is extracted for each image thus creating a new dataset of features that is used to train the classifier. The authors noted that although linear models are considered interpretable, it does not imply that they can provide good explanations by default. To generate good explanations, they identified three conditions to satisfy:

1. Identify the features that highly determine the prediction result.
2. Identify some other neighboring instances that share the same prediction output and at least one common explanation rule.
3. Identify the critical values of the most important features that would lead to a change in prediction.

Finally, they generated the explanations as presented in Figure 5.3. There are several formats such as visualization, graph diagrams and question-answers. Two sets of question-answers are proposed for two different levels of expertise. This system is a complete XIS as it contains every part of the system described in Figure 5.1 i.e. an explainable model trained with an adequate process and an explanation interface that engages some form of interaction thanks to the question-answer form.

Hendricks et al. [177, 178] designed a solution to generate textual explanations for a classification. These explanations are both image relevant and class relevant as they correctly identify features that are present in an image and are discriminative evidence of a class. A finegrained classifier is necessarily used as it is capable of detecting both the classes and sub-categories of these classes (e.g. a bird is the main class, a beak or a tail are sub-categories of a bird). To do so, several explanations are sampled based on the features of the main class. A score corresponding to the confidence that a feature is visible in the image is calculated for each feature. Then, the features with the highest score are used for the explanation. For instance, the class Scarlet Tanager (a species of bird) is known by the finegrained classifier to be red, have a black tail, a long beak and a black belly. The image only clearly shows that the bird is red and has a black tail, therefore, the generated explanation will be "This is a Scarlet Tanager because it is a red bird with a black tail" alongside bounding boxes of the corresponding features. However, most elements of this system use black-box models which may lead to unfaithful or biased explanations because of the lack of robustness of these models.

In summary, explainability methods for image classification are focused on feature attribution or saliency maps to extract visual explanations. These methods are controversial as they suffer from several issues such as unreliability and unfaithfulness. In addition, most methods employ black-box approaches to generate the explanations which exposes the explanations to the same issues e.g. a lack of robustness. Therefore, in the past years, scholars have started exploring alternatives to explain image classifiers. Notably, Pintelas et al. [55] created a complete XIS that requires the extraction of human understandable features that are also useful to train a machine

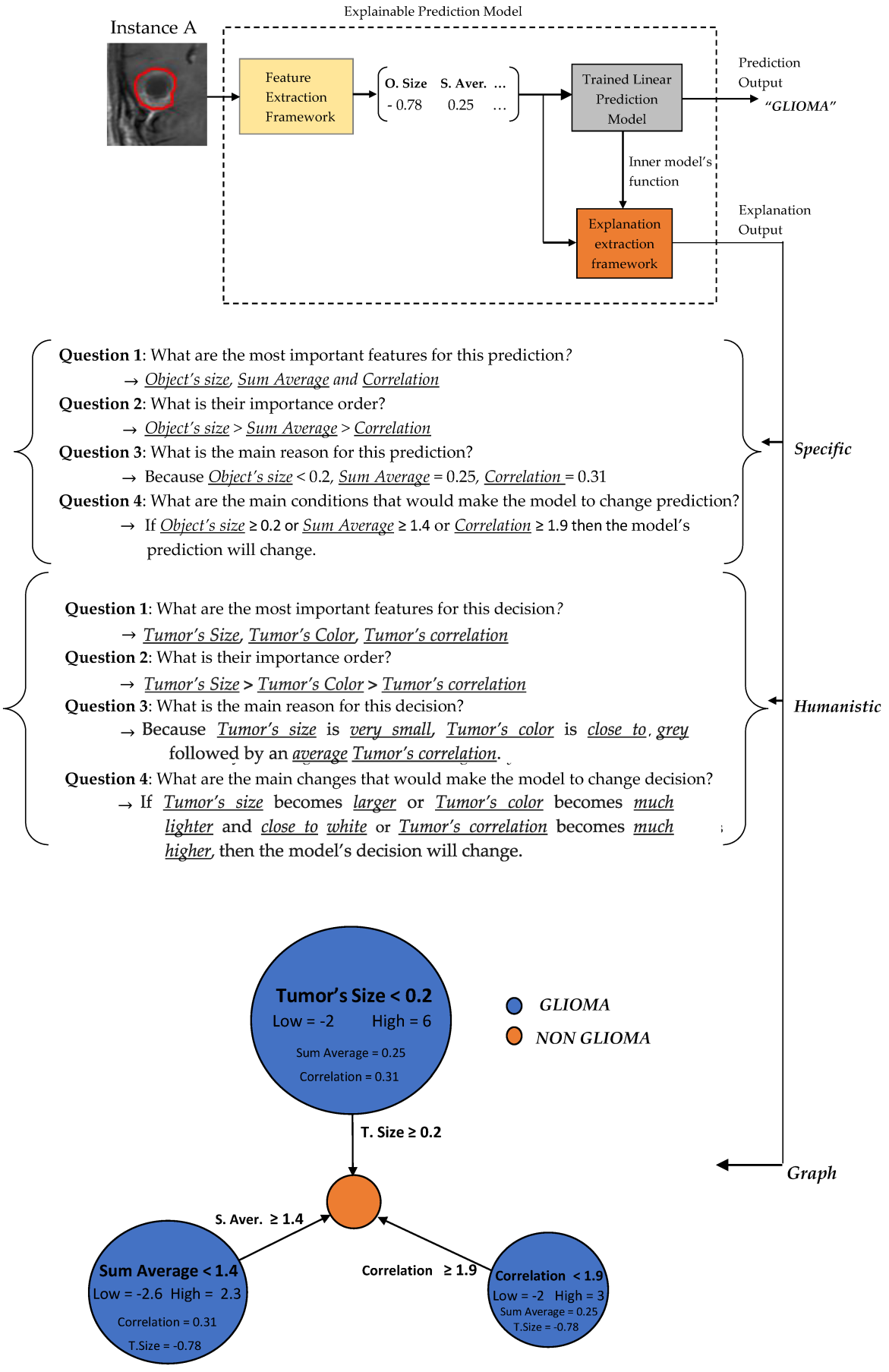


Figure 5.3: The explanation output from the method of Pintelas et al. [55].

learning model. This task requires manual intervention and both domain-knowledge and AI expertise to identify ideal features. Moreover, they use a linear binary classifier which limits the application to a binary problem. Likewise, high-performance predictions are not guaranteed because the chosen set of features and the limitations of the linear classifier.

The integration of domain-knowledge may be beneficial to current solutions. For instance, the set of features for the method of Pintelas et al. [55] may be dictated by the concepts of an ontology that describes certain classes. Similarly, the solution proposed by Hendricks et al. [178] necessitate a finegrained classifier to know the features associated to a class. A knowledge-base also contains this information but is less costly since it does not require to be trained.

5.1.3 Explainable neurosymbolic methods

Neurosymbolic models have seen a recent increase in interest. We have discussed in Chapter 3 that although these new models are prone to be used for explainable systems, few neurosymbolic models are actually designed for explainability. Harmelen and Teije [179] created a boxology of design patterns for systems that combine machine learning and knowledge representation models. Symbolic and numerical data are considered as inputs and outputs of these models. The proposed boxology discusses how the different models and types of data can be arranged to produce neurosymbolic models. In particular, three design patterns for explainable systems are described. The authors argued that explainable systems should output symbolic data as it is more amenable to crafting an explanation of the learning result than numerical data. The first design is a machine learning model that takes numerical data as input and outputs symbols. These symbols are fed to a knowledge representation model which crafts an explanation of the results of the machine learning model. The second design uses the same structure with the difference that the knowledge representation model exploits background knowledge in addition to the output of the machine learning model to create the explanation. Finally, the third design uses a knowledge representation for inspection of the behavior of the machine learning system. A knowledge representation model is given both the numerical input and output of the machine learning model to generate an explanation.

Several scholars promoted the use of Semantic Web Technologies to create explainable neurosymbolic methods [31, 82, 147]. Consequently, most of these methods employ ontologies jointly with machine learning. Some work utilize ontologies separately from the machine learning model to improve and explain the prediction. Geng et al. [56] used an ontology to generate explanations for predictions made with transfer learning or zero-shot learning. For instance, a model is trained to detect a cat and a cheetah. An image of a serval (i.e. an unseen class) which has a cat-like face and a cheetah-like body is given as input. The model is able to detect a serval and justify this prediction by manipulating the ontological relations between the predicted unseen class and the two known classes that have also been detected in the image. Pommelet and Lécué [57] described the application of ontologies for explainable object detection. They estimated the confidence scores of output classes based on their semantic description. For instance, the detection of a car in an image has a higher confidence score if properties of a car described in the ontologies are also detected in the image e.g. having wheels or being on a road. Marino et al. [58] proposed a framework for image classification that detects features in an image that correspond to elements of an ontology. Based on these features, a classification is done by finding the element in the ontology that is related to the most features. The classification is done through logical reasoning and can therefore be explained. We note that these methods echo the explanations proposed by Hendricks et al. [178] discussed previously, where the detected features of a class were used to generate explanations.

Ontologies are also employed to dictate the architecture of machine learning models. Phan et al. [180] used health ontologies to train and explain a model that predicts human behavior based on their activity in health social networks. Voogd et al. [181] introduced the Relational Concept Network (RCN) which is a network where symbolic concepts are connected together via edges that represent relations. Concepts and their relations are extracted from expert knowledge. Each

concept is attributed a model (from simple rules to neural networks) that outputs the concept's activation based on the activation of the other connected concepts. An input concept has its activation value set with external sources e.g. a sensor or human input. The activation values are then propagated through the network to obtain the activation of every concept. The final prediction determines the presence or absence of a concept that is explained by the activation value of every prior concept. The expert knowledge from which the concepts are extracted was entered manually. However it is clear that ontologies can be used instead.

Bourguin et al. [182] proposed a framework to design explainable classifiers that mixes both approaches i.e. use an ontology to either train a machine learning model or improve and explain a prediction. Indeed, their framework is divided into two parts: a deep learning segmentation model and an ontological classifier. The segmentation model is trained to extract the ontological features of an image. For instance, a pizza has several topping defined in the ontology. A dataset must be built where the position of the toppings are annotated. The segmentation model trained with this dataset is able to detect ontological features i.e. toppings of the pizza. Each pixel is mapped to an assertion and given to the second part of the framework. This part classifies the object in the image with the structure of the ontology and the observed assertions. Yet, the full power of a classic logical reasoner is not needed to make the classification. Therefore, they proposed a method named OntoClassifier that automatically generates a graph of tensors that are interconnected according to the class definitions in the ontology. This graph of tensors allows the classification of the object in the image while retaining the possibility to trace back the reasons of this classification. The advantage of the OntoClassifier is that it can be directly included in the machine learning pipeline and is much faster than a logical reasoner.

Most of the studied neurosymbolic explainable methods use a black-box at some point, mostly to detect features in an image. The RCN method [181] addresses this problem by proposing a modular approach where the choice of machine learning model is free and compatible with interpretable models. Current neurosymbolic methods rarely discuss the generation and presentation of the explanations. We observe that the provided explanations are mostly designed for domain and AI experts, not for the end-user. Additionally, these methods rely on the predictions of a machine learning model without implementing any fail-safe in case the predictions are wrong. Alirezaie et al. [183] presented a symbolic approach to explain misclassifications of an image classifier. They apply spatial reasoning alongside a domain-specific ontology to determine why some regions of the image were misclassified. However, this method needs the true label of the image to detect that an error occurred and identify the misclassified regions. This method is useful for debugging the machine learning model. Therefore, the main limitations of current explainable neurosymbolic methods are the absence of error detection, the lack of exploration on how to generate explanations and the reliance on black-box models to make the predictions.

5.1.4 Explanation interfaces

Explanations are an interactive process. Scholars agree that the presentation format and the interactivity of the explanations are crucial to make good explanations [11, 15, 34, 55]. Consequently, several explanation interfaces have been created to present explanations in an interactive way. Moreover, guidelines and principles on the design of explanation interfaces are discussed to help practitioners bring explainability methods to AI products.

Liao et al. [184] interviewed practitioners working on AI products to find gaps between the current research on XAI and the practices to bring XAI products to the end-user. They identified user needs to understand AI systems and proposed a question bank to represent them. We note that the content of this question bank aligns with the several goals of XAI discussed in Chapter 2. This study shows that there is little to no research or shared practices on how to design user-friendly XAI apps. Chromik and Butz [185] reviewed and proposed principles on the design of user interfaces specific to explainability. They defined an explanation user interface (XUI) as the sum of outputs of an XAI process that the user can directly interact with. The notion of *raw explanations* was also introduced which corresponds to the direct output of explainability techniques as described by AI

researchers. An XUI takes these *raw explanations* and adapts their presentation in a user-friendly manner. Then, the authors described the different goals of interactions between humans and AI along with XUI architectures that achieve them. Two types of XUIs are defined: explanatory XUIs aim to convey a single explanation whereas exploratory XUIs let users freely explore the model behavior. Exploratory XUIs are more effective as the user can explore the interface to get different explanations on different levels, i.e. global explanations to understand how the system works or local explanations to understand a particular prediction. Exploratory XUIs enable the application of the practices recommended by Liao et al. [184]. Additionally, Amershi et al. [186] compiled the work of the human-computer interaction (HCI) field to create and evaluate 18 generally applicable guidelines for human-AI interaction. We observe that there are ongoing efforts from the research community to help AI and design practitioners to implement explainability in their products.

Chromik and Butz [185] mentioned a lack of research on the design of human-centered XUIs. Nevertheless, several interfaces have been proposed in the recent years:

- The What-If tool [187] is made for AI experts to explore and diagnose their data and models. In this sense, it follows the guidelines from Liao et al. [184] as it addresses identified user needs, where the users are AI experts. Data visualizations, counterfactual explanations and fairness metrics for machine learning models are among the main features of this interface.

- The Neuroscope interface [47] provides a graphical user interface to present the results of a variety of XAI methods for image classification and segmentation, mostly saliency maps (see Section 5.1.2). However, we have seen that this class of explanations is not reliable or faithful to the actual behavior of a model. Moreover, the explanation presentation is not user-friendly and is useful mainly for AI experts to debug and improve their models.

- Several XUIs have been created by implementing popular XAI methods in a single interface ([188, 189, 190]). Still, these XUIs rarely transform the output of these methods (i.e. *raw explanations* [185]) into a user-friendly presentation. The user may be overwhelmed by the variety of explanations proposed without guidance.

- Jin et al. [191] presented the EUCA framework (End-User-Centered Explainable AI) to alleviate the issues from the previous group of XUIs. This framework enables AI practitioners to easily design an XAI interface with over 12 forms of explanations. The explanations are intended for end-users with different roles, goals and levels of expertise. EUCA functions with "prototyping cards" that instantiate different explanation forms. These cards are selected and customized to perfectly fit the needs of users via a cooperative design between end-users, stakeholders and practitioners. The authors observed that there rarely is an overlap between experts in AI and experts in HCI despite the need for both domains in XAI.

In the past years, we noticed an increase in the development of explanation interfaces that apply previously identified principles and guidelines. It has been discussed that there is a lack of collaboration between the AI and HCI communities to design explainable intelligent. In the current situation, both communities have been working separately. As a result, the state-of-the-art XUIs proposed by the HCI community only support machine learning models and popular post hoc explanations (e.g. LIME, SHAP...). Although HCI researchers are designing flexible interfaces that facilitate the implementation of new XAI algorithms, it is unclear whether symbolic or neurosymbolic models and their related explanations could also be easily implemented into them.

5.1.5 Discussion

To summarize this literature review, we identified several limitations in the XAI field to explain image classifiers. First, error detection techniques are mostly ignored by the XAI community although the outcome of these techniques is a gain in safety and trustworthiness of AI systems. Indeed, these techniques allow an AI system to say "I am not sure" when the prediction is uncertain. Moreover, the existing methods made for XAI do not provide satisfactory explanations.

Secondly, the current research in XAI methods for image classifiers is focused on the creation of saliency maps which may be unreliable and unfaithful to the actual behavior of the image classifier. Recently, scholars developed promising alternatives that generate several types of explanations by using interpretable models. We observed that these alternatives could be improved by using domain-knowledge to guide some important choices such as the choice of interpretable features used for a prediction. Hence, we explored neurosymbolic approaches for classification. A majority of these approaches use ontologies to either improve a prediction or determine the architecture of a machine learning model. A common idea is to detect the presence or absence of certain concepts of an ontology with a machine learning model and then use these predictions as input of a reasoner to make the classification. However, black box machine learning models are still employed to make some predictions meaning that undetected errors can occur and additional post hoc explanations are required. Additionally, the explanations are usually designed for AI experts and not laypersons or stakeholders. Finally, we studied the design of explanation interfaces. We noted a recent interest by the HCI community in the design of explainable user interfaces or XUIs. The main takeaway is that the XAI researchers should collaborate with the HCI community to design XUIs that are adapted to the end-users needs while applying the most adequate XAI solutions. We reviewed existing XUIs and noted that none of them implement neurosymbolic methods, but rather focus on popular XAI algorithms limited to explaining only machine learning systems.

5.2 OBIC: explainable ontology-based image classifier

We designed an explainable ontology-based image classifier (OBIC) architecture. This classifier is agnostic of the machine learning model thus allowing the use of interpretable models. An explainable error detection solution is proposed to increase the safety and trustworthiness of this system. The goal of this system is to require minimal efforts from the AI practitioners to implement it. Specifically, it allows the reuse of existing ontologies and classifiers with little to no additional work. Example 5.1 will be used throughout this chapter to illustrate each step of OBIC.

Example 5.1 In this example, we assume that we are given a dataset of images of furniture and an ontology that describes furniture. In this ontology, two object properties are defined that correspond to visual properties of furniture:

- `:hasMaterial` corresponds to the material of a piece of furniture e.g. wood or metal.
- `:hasColor` corresponds to the color of a piece of furniture such as brown, red or yellow.

We focus on the definition of the `WoodenChair` class which is a subclass of `Chair` and is defined as follows in Algorithm 5.1.

Algorithm 5.1 Class definition of `WoodenChair`, with the OWL2 Manchester Syntax [143]

Class: `WoodenChair`
SubClassOf: `Chair`
SubClassOf: `:hasMaterial` some `Wood`
SubClassOf: not `:hasMaterial` `Metal`
SubClassOf: `:hasColor` some `Brown`
SubClassOf: not `:hasColor` some `Yellow`

The functioning of OBIC is illustrated in Figure 5.4. It is similar to the approaches seen in [55, 181, 182]. First, machine learning models are trained to detect the main class and the presence or absence of a set of concepts from the ontology's T-Box. Specifically, one machine learning model referred to as the *global classifier* is trained to carry out the main classification task i.e. it corresponds to a classical machine learning pipeline. The other models are trained to detect whether an object property is present in the image and if so, determine the class of the object. Then, an individual representing the image is created. The class predicted by the *global classifier* is given to the individual in the form of a `ClassAssertion`. Similarly, the detected object properties are also added in the form of `ObjectPropertyAssertions`. Finally, the individual is added to the ontology's A-Box and a logical reasoner is run to check the consistency of this individual. This consistency check acts as an explainable error detection system since the reasoning uses logic and human concepts that can be explained.

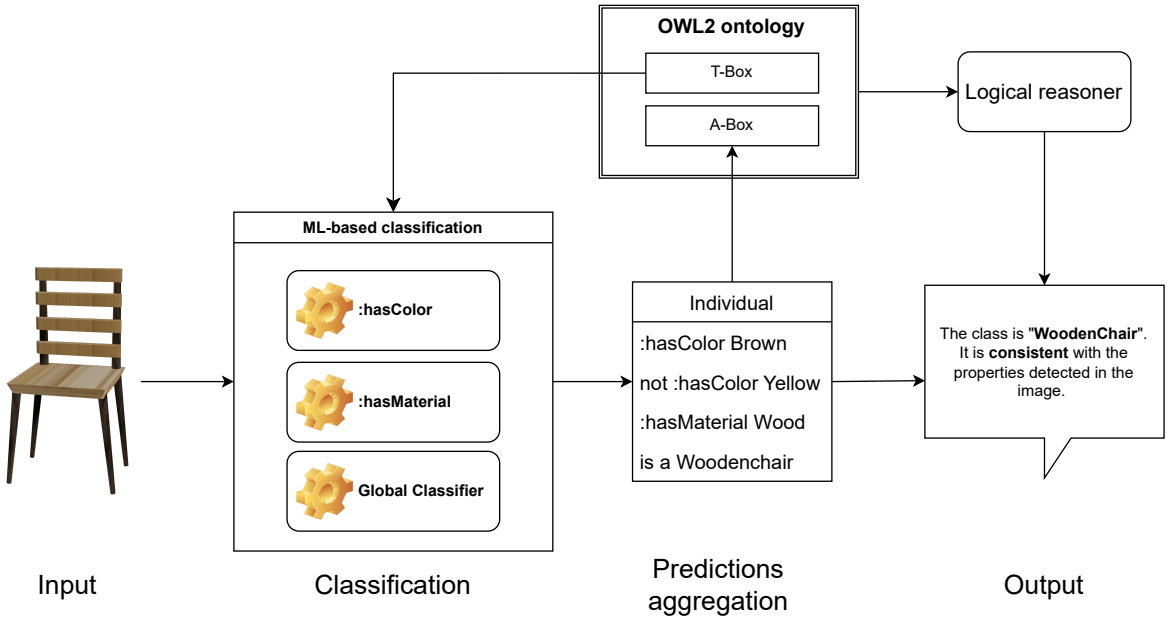


Figure 5.4: Diagram of the functioning of OBIC

The OBIC framework uses the design pattern of an *explainable learning system with background knowledge* proposed in [179] and illustrated in Figure 5.5 where "ML" corresponds to an inductive component, "KR" corresponds to a deductive component, "data" is a numerical input/output format and "sym" a symbolic input/output format. Data in the form of an image is fed to machine learning classifiers. The output of the classifiers is symbolic since it is an individual containing concepts and properties detected by the classifiers. The deductive component (i.e. the ontology) uses background knowledge and the output of the inductive component to detect errors and present the prediction along with explanations.

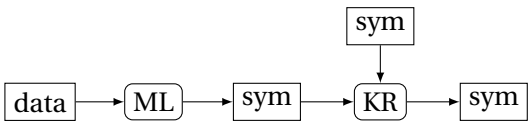


Figure 5.5: Design pattern from [179] used by OBIC

First, we discuss the requirements on the ontology necessary to implement OBIC. Then, we present the building and training of the machine learning models based on the ontology. Finally, we describe the inference phase of OBIC, particularly how to extract assertions from the output of the classifiers.

5.2.1 Ontology requirements

The OBIC framework is tailored for image classification. It is assumed that a dataset containing images of each class is available. An ontology that formally describes this dataset is required, hence each class of the dataset should be defined in the ontology. The class definitions of the ontology should include object properties that describe observable characteristics. An observable characteristic is a characteristic of a class that can be observed in the dataset. For instance, characteristics of furniture that can be seen in an image are the material or the color as shown in Example 5.1.

An abstract object property named `:observableProperty` is introduced, which corresponds to any observable characteristic in the data. Object properties that describe an observable characteristic are defined as `SubPropertyOf :observableProperty`. The domain of an observable property contains the set of classes present in the dataset. The range of an observable property can be any set of classes defined in the ontology. Then, the definition of each dataset class in the ontology should contain at least one restriction with an observable property. Algorithm 5.2 provides the definition of the property `:hasMaterial` used in Example 5.1 to define a dataset class in the ontology.

Algorithm 5.2 Example of observable property definition, using OWL2 Manchester Syntax [143]

ObjectProperty: `:hasMaterial`
Domain: `Furniture`
Range: `Material`
SubPropertyOf: `:observableProperty`

To get the best results from OBIC, the class definitions should be as exhaustive as possible. Indeed, inconsistencies are detected by a logical reasoner that uses the open-world assumption. Therefore, definitions with negative restrictions are encouraged in order to help the reasoner find inconsistencies. Moreover, class definitions are used in the training phase to relabel the dataset and improve the accuracy of each classifier. Still, observable characteristics are not necessarily present in every image. Many factors could hide some characteristics which could lead to a misclassification later. Although class definitions should be as exhaustive as possible, they should also take into account the probability that a characteristic may be hidden in the definition. A characteristic that is likely to be hidden in most images can be included in the definition but not as a necessary condition. For instance, the presence of hangers inside a wardrobe should not be necessary to classify a wardrobe, as it is highly probable that clothes may hide the hangers or that the wardrobe is closed.

5.2.2 Training phase

Our proposed method creates one machine learning model named *global classifier* to find the main class of an image. This model is built and trained with the available dataset and independently of the ontology. In parallel, multiple models are built and trained to detect each observable characteristic (i.e. sub-properties of `:observableProperty`) defined in the ontology. In the following, we refer to these models as *property classifiers* or *property models*. Their goal is to determine the assertions identified in the image. We refrained from using a segmentation model or a finegrained classifier to detect the observable characteristics because these tasks would encourage the use of black-box models and necessitate datasets specifically built to that end. Moreover, the models are trained separately to prevent any correlation between output classes that would produce skewed predictions. For example, given one monolithic model capable classifying every property at once, the prediction of a wooden chair would systematically imply the detection of wood. Indeed, the two classes are highly correlated and this is what the model would learn. However, the goal of a property classifier is to be able to generalize concepts i.e. when an image of an unseen class is given, the property classifier should still be able to accurately detect the presence and type of a property. Therefore, the classifiers are trained separately which increases the cost of

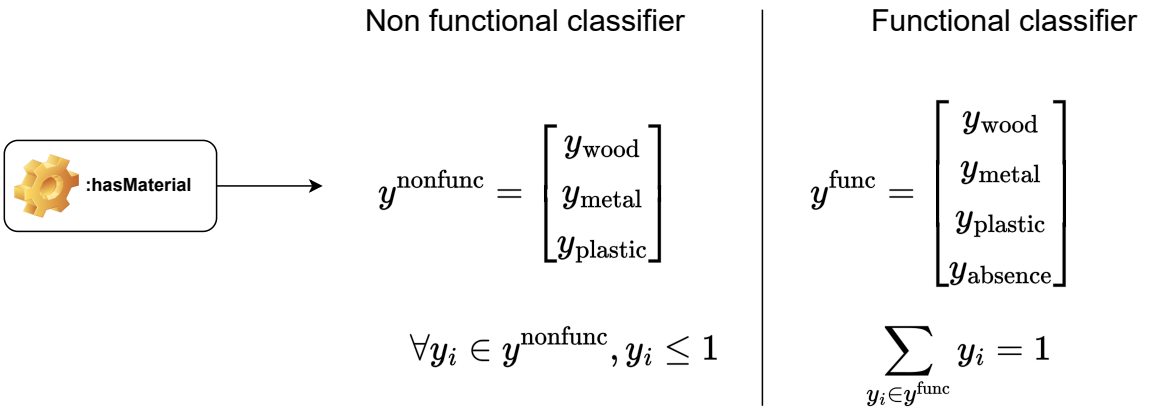


Figure 5.6: Output of the `:hasMechanism` classifier when the classes `Wood`, `Metal` and `Plastic` are in its range.

training but should improve their performance. Two problems emerge from this solution: how to build the machine learning models and where to find adequate datasets to train them.

Building the classifiers

The purpose of a property model is to predict whether the property is present in an image and if so, predict the class of the object of the corresponding assertion. We observe that in some cases, there might be multiple assertions with the same property e.g. an object may have multiple colors or materials. In other cases, only one assertion with the same property is possible e.g. a person only has one date of birth. The architecture of a property classifier depends on its classification task. There are two types of classification task: multi-class classification is the task of classifying an element into one of several classes while multi-label classification is the task of attributing multiple classes to one element. In multi-class classifiers, the output's sum equals 1 and the class with the highest score is selected. The *global classifier* is always a multi-class classifier because the dataset is assumed to attribute a single label per image. Regarding property classifiers, the two classification problems can appear. In cases where an individual can have multiple assertions with the same object property, a multi-label classifier is used. Finally, some object properties are defined as `functional` i.e. for each individual x, there can be at most one distinct individual y such that x is connected to y by a functional property [141]. Hence, the property model for a functional object property is a multi-class classifier.

The output of a property classifier is a one-hot-encoded vector where each element corresponds to a class in the range of the observable property. In the case of functional properties, the sum of each element of the output is necessarily 1. Consequently, an additional element is added to allow the model to predict the absence of the property. For instance, let the classes `Wood`, `Metal` and `Plastic` be the range of the property `:hasMaterial`. The output of the classifier of `:hasMaterial` is a vector with 3 items. Figure 5.6 shows the output format of this classifier in this example. The expected output of a chair made of wood and of metal is $(1,1,0)$ which means that the assertions $(chair - \texttt{:hasMaterial} - Wood)$ and $(chair - \texttt{:hasMaterial} - Metal)$ are detected in the image. Similarly, the expected output of an image showing a chair with no material (e.g. screenshot of a 3D-editing software) is $(0,0,0)$. If no class is detected, then it implies the absence of this property in the individual. If the property `:hasMaterial` were functional, an object could only have a single material. Hence, a label for the absence of the property is added at the end of the output vector, as illustrated by y^{func} in Figure 5.6. The output for a chair made of plastic would become $(0,0,1,0)$. For a chair with no material, it would be $(0,0,0,1)$ to ensure that the sum of each output is 1.

The selection of the classes represented in the output of a property classifier has not yet been discussed. There are three alternatives to select these classes that are all subclasses of the prop-

erty's range.

1. The first is to select all the classes of the hierarchy tree where the range is the root and the `SubClassOf` relation links classes together. This selection gives the classifier the possibility to detect an abstract class meaning that it could handle cases where the actual class is unknown but is part of a certain family e.g. if a certain color is not defined in the ontology, the classifier has the possibility to say that a color was detected without specifying which color. However, this would also create correlations in the output since when one class is detected, all its parents class should also be detected. We discussed earlier that correlations in the output are to be avoided as they hinder the performance and generalization power of the model.

2. The second alternative is to select the leaves of the hierarchy tree. It avoids the problem of correlation but imposes challenging requirements on the dataset. Indeed, the ontology may define very specific classes e.g. several subclasses that define types of wood such as oak or walnut. Thus, the dataset should contain a sufficient number of occurrences of the necessary labels to detect these specific classes.

3. The third alternative consists in manually selecting the classes based on the dataset. Doing so is equivalent to the third alternative, which consists in manually selecting the classes based on the dataset. This selection is ideal as it prevents every drawback from the two previous alternatives. Nevertheless, it comes at the expense of automation and requires additional work to implement OBIC.

In the implementation of OBIC, we chose the second alternative (i.e. the leaf classes) to retain the automation. Still, we made sure that the dataset and ontology definitions were in line to mitigate the drawbacks of this choice.

Data labeling

Training the property models requires a dataset with appropriate labels. We assume that the only available data is the dataset with the labels for the *global classifier*. Therefore, a method to relabel this dataset is proposed, based on the class definitions in the ontology.

Figure 5.7 shows the process of relabeling the dataset. The label of an image indicates which main class is present. Based on this information, we query the definition of this class in the ontology. Specifically, we analyze object property restrictions in the class definitions and extract the target class of these restrictions. The target classes of the restrictions are the new labels for the image. For instance, a `WoodenChair` is defined as a subclass of `:hasMaterial` some `Wood`, therefore an image of a wooden chair is also an image of wood. Hence, a new dataset can be created for each property model by using this information to infer the labels.

The application of the open-world assumption raises the question of how to convert this information into relevant values for the labels. Although positive and negative object property restrictions explicitly say that a class is respectively present or absent in the image, the absence of such restriction does not imply that other classes are not present. In other words, the definition of `WoodenChair` explicitly says that the material wood is present in the image. We have no information concerning the presence of other materials such as metal or plastic. The definition does not exclude the possibility that some wooden chairs may contain these other materials. One-hot-encoded labels for explicitly present or absent classes are respectively 1 or 0, however there is no ideal value for unmentioned classes. Setting the value of unmentioned classes to 1 is not a good choice since it would imply that a class is present even though there is no guarantee of its presence. A value of 0 may lead to a lower capacity of the model to detect a class. A value in-between seems adequate, especially a value of 0.5 which corresponds to the usual threshold to decide whether a class is present or absent in multi-label tasks. Yet, using such value could encourage the output of the class to remain around 0.5. The inference phase of OBIC exploits the output value as a probability of the presence of the class. Therefore, having output values around 0.5 is not ideal. The

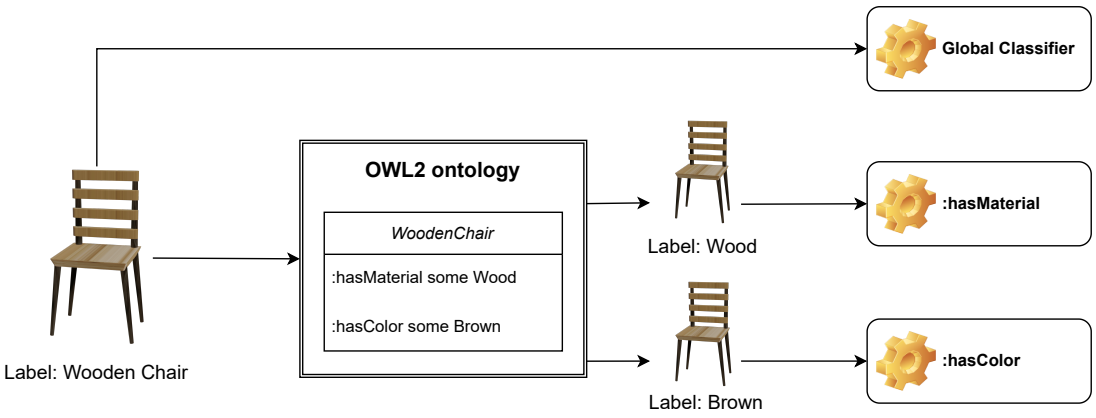


Figure 5.7: Illustration of the labeling and training process.

choice of this value depends on the dataset and the ontology. A value of 0 or 1 could lead to errors, though it is very common in machine learning to have some errors in the labels of the dataset. This problem only appears for multi-label classification task. Hence, we chose to set the default value of unmentioned classes to be 0, in order to remain consistent with multi-class classification in which the value of the other classes in the output are set to 0.

This process of automatically relabeling a dataset based on domain-knowledge extracted from an ontology has advantages outside the scope of explainability. It enables the reuse of datasets built for different tasks to generate a new dataset designed to detect a common concept. For example, it is possible to combine a dataset of furniture with datasets of cars and forests to create a dataset of materials. All of these datasets contain objects that have materials such as wood or metal. Although these materials are in totally different contexts, the concept of material remains the same. Hence, our proposed process to relabel a dataset based on concepts of an ontology can be employed to relabel and combine these datasets. Moreover, doing so may improve the ability of a machine learning model to generalize a concept since the data points come from a variety of sources. Similarly, this approach facilitates the addition of new classes and new properties in the ontology. When a new property is added, there is no need to retrain every model as property models are created independently of the others. Likewise, the addition of new classes and new data is automatically handled by OBIC. Still, in this case, the property models may need to be retrained with the new data to improve their performance.

5.2.3 Inference phase

The inference phase inputs an image to every trained classifier, gathers the outputs to create an individual that is then added to the ontology's A-Box as illustrated in Figure 5.4. The output of each classifier is a vector y where each element $y_i \in [0,1]$ represents the probability that a class is present in the image. The *global classifier* predicts the main class of the image, which is the class with the highest probability in the output. This prediction is added to the individual with a `ClassAssertion`. Then, the output from each property classifier is translated into `ObjectPropertyAssertions` with the individual as subject. This translation raises the problem of determining thresholds that decide the presence or absence of an assertion. This problem is similar to the issue of setting the value of the label in the case of unmentioned classes. In multi-label classification, the common practice is to say that an output greater than 0.5 means that the class is present and conversely, lower than 0.5 means that it is absent. However, as we observed in the review of error detection methods in Section 5.1.1, we want our models to be able to say "I do not know" when a prediction is uncertain. For each class in the output vector, we study the probability or confidence of the presence of this class. We identify three cases:

Explicit presence The probability that the class is present is high, which leads to adding the corresponding assertion to the individual.

Explicit absence The probability that the class is present is low, which leads to adding the corresponding negative assertion to the individual.

Uncertain presence The probability that the class is present is near 50%, meaning that the presence or absence of this class is uncertain. To handle this case, we take advantage of the open-world assumption by not adding any assertion to the individual. Hence, neither the presence or absence of the class is assumed when checking the consistency of the individual.

Two thresholds should be fixed to determine what is "high enough" and "low enough". We propose two parameters called $threshold^+$ and $threshold^-$ such that:

$$0 \leq threshold^- \leq threshold^+ \leq 1$$

Output values lower than $threshold^-$ fall under the explicit absence case, values greater than $threshold^+$ fall under the explicit presence case and values in-between the two thresholds are considered uncertain. We add the constraint that the sum of both thresholds equals to 1 i.e. $threshold^+ + threshold^- = 1$. Therefore, the process is the same for multi-label and multi-class classification. Indeed, the sum of all output values is equal to 1 in multi-class classification. Thus, there can only be one class that has a probability greater than $threshold^+$. If no value is greater than $threshold^+$, then we consider that there is too much uncertainty and nothing is considered explicitly present. Still, several classes may fall under the explicit absence. We note that the common practice for multi-label classification (i.e. a threshold of 0.5) is a particular case of this framework, where the choice of thresholds is $threshold^+ = threshold^- = 0.5$. The impact of the thresholds values on the behavior of OBIC is unknown. Thus, we do not yet provide a methodology to select the best values. The impact of the thresholds on the behavior of OBIC is explored and discussed in Chapter 7.

Algorithm 5.3 describes the process of handling the predictions of each classifier and adding the corresponding assertions to a new individual that represents the object in the image. We can observe that the case of functional object properties is handled similarly to non functional properties. The output of this function is the individual to be added to the ontology. The error detection is straightforward, a logical reasoner is called to check the consistency of the ontology with the new individual. It is assumed that the ontology is consistent before adding this individual, therefore any inconsistency is caused by the individual. The detection of faulty assertions (i.e. assertions that provoke an inconsistency) is done by iteratively adding one assertion at a time and testing the consistency of the ontology with the new assertion. The list of faulty assertions gives information on which classifier may have been wrong. We will further discuss the analysis of the errors in Section 5.3, along with the presentation of the output and the explanations.

The proposed OBIC framework automates the creation of an explainable image classifier with minimal effort from the AI practitioner. A dataset along with an ontology that defines the classes of the dataset are needed. Both may already exist and the ontology would need a minimal amount of manual tweaking to meet the requirements necessary for OBIC. From these assets, classifiers for each observable characteristic are built and trained thanks to the automatic relabeling process. Moreover, any machine learning model can be used to create the property classifiers. This allows the choice of any classifier with the adequate characteristic for the task e.g. interpretability or a high accuracy. Then, an individual is created by translating the output of each classifier into assertions. The direct output of the classifiers are seen as the confidence score that a class is present. In the literature review, we have seen that this practice has conceptual drawbacks. Some different confidence scores can be applied such as the *trust score* [157] without modifying the functioning of OBIC. Finally, a logical reasoner is run to detect inconsistencies in the individual which are provoked by errors in the classifiers. This framework is similar to the one proposed by Bourguin et

Algorithm 5.3 Inference algorithm using the OWL2 functional syntax [141].

function INFER($x, observablePropertiesSet, threshold^+, threshold^-$) ▷ Returns an ontology individual with the assertions detected by the classifiers.
  $class \leftarrow \arg\max globalClassifier(x)$ ▷ Get the class predicted by the *global classifier*
  $Declaration(NamedIndividual(indiv))$
  $ClassAssertion(class, indiv)$
  for all $property$ in $observablePropertiesSet$ **do**
    $y \leftarrow propertyClassifier(property, x)$ ▷ y is the output of the property classifier
    for $i \leftarrow 0, length(y)$ **do**
      $Declaration\big(NamedIndividual(target_i)\big)$
      $ClassAssertion\big(class_i, target_i\big)$ ▷ Create an individual of the i-th class in the labels for $property$
      if $y_i \geq threshold^+$ **then**
        $ObjectPropertyAssertion(property, indiv, target_i)$
      else if $y_i \leq threshold^-$ **then**
        $ClassAssertion(ObjectComplementOf(ObjectSomeValuesFrom(property, class_i)), indiv)$
      end if
    end for
  end for
  return $indiv$
end function

al. [182] and provides a solution to the problem of manually selecting the features to detect. The feature extraction step in OBIC is fully automated based on the ontology. Despite the automation of this part and our attempt to minimize human intervention, it is still required to adapt the ontology and choose adequate thresholds. This lack of full automation was regretted by Tiddi and Schlobach [86].

5.3 Explanations with OBIC

In the previous section, we introduced the design of OBIC which is an explainable model based on ontologies. The goal of using an ontology is to ensure that the predictions exploit human understandable concepts. Furthermore, the logical reasoning applied to check for the consistency of the prediction enables the generation of faithful explanations as the causes that led to the consistency can be traced back. In this section, we explore the explanations that can be extracted from OBIC to explain both the predictions and the error detection. Then, we propose a design for an explanation interface that presents and explains the outcome of OBIC, in order to complete the XIS.

5.3.1 Extraction of the explanations

We have designed an explainable model for image classification and described the corresponding training process, as illustrated in the DARPA's architecture for an explainable system [15]. In order to make a complete XIS, an explanation interface has to be created that extracts information from the explainable model to present and explain the results to the user. The information that can be extracted from OBIC is the input image, the output of each classifier, the thresholds, the assertions of the individual, the consistency of the prediction, the faulty assertions and the ontology. We observe that there is no available information concerning the functioning of the machine learning models. Indeed, any machine learning model can be employed with OBIC. Therefore, we are not able to explain the reasons for the class prediction since the *global classifier* is responsible for the class prediction and not the ontology. The choice of explainability method to explain a prediction

is usually motivated by the type of machine learning model (e.g. interpretable models, neural networks). We intentionally avoid making any default choice about the explanation of the prediction in order to maximize the compatibility of OBIC with any system capable of classifying an image. Nevertheless, the available information enables the explanation of the consistency check.

The proposed explanation interface displays the results of the system and explains the reasons for the inconsistency. The design of this explainable user interface (XUI) and the explanations are guided by the principles and guidelines discussed in Section 5.1.4 as well as the guidelines for a Responsible AI discussed in Chapters 2 and 3. In this XUI, we make available every information concerning the prediction to ensure transparency and human oversight. Moreover, the error detection step that improves the safety of the system is presented and explained alongside the prediction to help the user decide whether to trust the prediction. Similarly, user-centered explanations are strongly advocated by scholars. We distinguish three levels of expertise, as discussed in mentioned in Chapter 3: AI experts, domain experts and laypersons. We build our explanations and XUI starting from simple and comprehensible information for a layperson and extend it to suit the needs of domain and AI experts. We note that the comprehensibility of an information varies from user to user, therefore this notion is biased and would require a user-study to ensure that the proposed explanations are actually comprehensible for laypersons.

The first step is to display the input image and results of the prediction as it is the main task of the system. The consistency or inconsistency of the class prediction is presented alongside the prediction to efficiently warn the user when the prediction cannot be trusted. Then, we expect that the user will ask two questions: "why was this class predicted" and "why can or can't it be trusted?". Although we have mentioned that we do not have the necessary information to answer the first question, the consistency can be explained with the available information. The assertions made on the individual are listed and the inconsistent assertions are highlighted. In other words, we present the symbolic output of the classifiers as described in [179]. Unfortunately, this explanation may not be comprehensible by laypersons as it directly uses the names from the ontology. Indeed, ontologies usually employ a technical vocabulary that is not adapted for laypersons, rendering the explanations difficult to understand for them.

Identifying the faulty assertions provides hints to determine which classifiers made mistakes. The mistakes were made either by the *global classifier* or by the property classifiers that led to the faulty assertion. The proportion of property classifiers that generated faulty assertions is a good indication of the source of the wrong predictions. When a majority of property classifiers led to the generation faulty assertions, it is probable that the *global classifier* made the mistake. Conversely, when a minority of property classifiers generated faulty assertions, then these classifiers are probably mistaken. This explanation encourages the user to further analyze the result to identify the wrong output and decide whether to trust the prediction. We expect the user to analyze the input image, looking for the presence or absence of the faulty assertion. Although this visual analysis generally requires domain knowledge, some concepts and properties may be identifiable by laypersons. For instance, a layperson that has no knowledge on furniture could still identify characteristics such as the material or color. If the presence of a faulty assertion is verified in the image, then it is the *global classifier* that made a mistake. Inversely, if a user cannot observe a faulty assertion, then it is the property classifier that is wrong. The ability of OBIC to accurately detect errors is evaluated in Chapter 7.

For more advanced users, we show the results of each output along with the thresholds values and a description of the process of generating assertions from the classifiers' output. This additional information enables the user to further analyze the reasons that led to an assertion in the final result. A domain expert can assess the quality of a classifier by studying the output for each class compared to the input. Similarly, the AI expert may use this information to debug and improve the system. Moreover, providing this information is necessary to make the system as transparent as possible.

5.3.2 Design of the explanation interface

We have discussed the results and explanations to present in the interface. The following proposition of a XUI is designed to display these components while satisfying several goals. The XAI and HCI communities encourage the design of user-centered explanations and interfaces i.e. create explanations and interfaces accessible to any user. To achieve this goal, information and explanations are presented in both visual and textual form when possible. Similarly, the interface is implemented to be compatible with a wide variety of devices. We adopted a modular approach to facilitate the addition of new explanation techniques. Considering our lack of expertise in HCI, the proposed interface is solely a proof-of-concept to demonstrate the possibilities offered by the OBIC framework.

The interface is shown in Figure 5.8 on the example of furniture classification. It is divided into three sections:

Input image The input image is displayed on the upper left of the interface. Showing the input is important as it allows the users to compare the predictions of the system with the input in order to detect any problem and assess the quality of the predictions and explanations.

Prediction and consistency The section on the upper right of the interface displays the predicted class in a large font to draw the attention of the user. When the prediction is consistent, the text is in green otherwise it is in red. This color code can be modified to better suit some users (e.g. colorblind persons may struggle with these colors). Underneath the class prediction, a sentence describes in a textual format the result and the consistency of the prediction. Then, the list of assertions is shown; the faulty assertions are separated from the other assertions. The current form uses keywords directly extracted from the ontology which may hinder the comprehensibility of this part for laypersons.

Output of the classifiers The output of each classifier is presented at the bottom of the interface using tabs. Each tab reveals the output of one classifier in a bar graph. This graph is color-coded to carry two information: the resulting assertion (i.e. explicit presence, explicit absence or uncertain presence as described in Section 5.2.3) and the consistency of the assertion. On the right of the graph, a legend explains how the assertions are decided and provides the values of the thresholds. The label of each class is also color-coded to represent the consistency of the resulting assertion. For instance, the prediction of `:hasMaterial Metal` in Figure 5.8b is inconsistent with the prediction of a wooden chair. Consequently, the label "Metal" for the results of the `:hasMaterial` classifier is in red while the other consistent labels are in green.

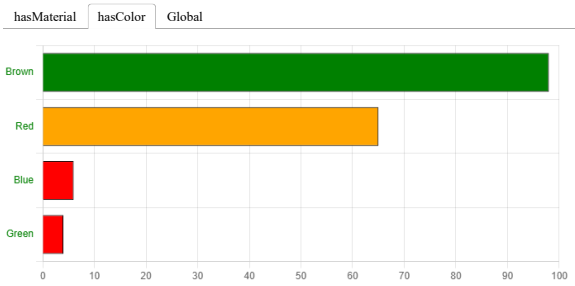
In its current state, the proposed XUI presents the results and consistency of the the system. It demonstrates that it is possible to exhaustively display the information made available by the OBIC framework leading to a transparent system. Nevertheless, this prototype has several shortcomings that may be improved by collaborating with the HCI community. Namely, the color code is cluttered and lacks clarity, the naming scheme and presentation of the assertions are too technical and not adequate for laypersons. Moreover, we identified that exploratory XUIs are ideal to enable personalized explanations yet the single interaction in this XUI is the exploration of the results of the classifiers via a tab system. Lastly, the explanation for the consistency is limited as there is no XAI solution to explain the consistency of an ontology to laypersons.

Chromik and Butz [185] specifically mention that humans gain understanding in many ways, meaning that several forms of explanations should be available to explain a same event. Thus, the current interface could benefit from additional explanations. A global explanation that shows the functioning of the OBIC framework through text, diagrams and examples may be beneficial to better understand the local explanations provided in the current XUI. Likewise, examples that illustrate the functioning of the framework and its limitations could improve the transparency and increase trust from the users. Interactive explanations should be the main focus to improve the current XUI. For instance, the threshold system may be interactive by letting the user modify the

WoodenChair

The model saw a **WoodenChair** and it is **consistent** with the observed properties.

The following properties were observed:
- hasMaterial Wood
- Not hasMaterial Metal
- Not hasMaterial Plastic
- hasColor Brown
- Not hasColor Blue
- Not hasColor Green



- Green bar: the system has decided the property is present because it has a probability higher than 70 %.
- Red bar: the system has decided the property is absent because it has a probability lower than 30 %.
- Orange bar: the system could not decide about the presence/absence of this property.

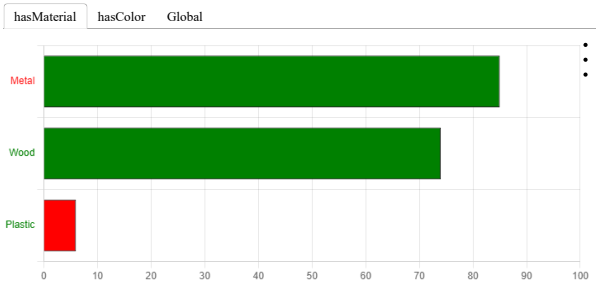
(a) A consistent case

WoodenChair

The model saw a **WoodenChair** and it is **not consistent** with the observed properties.

The following properties were observed:
- hasMaterial Wood
- Not hasMaterial Plastic
- hasColor Brown
- Not hasColor Blue
- Not hasColor Green

The following properties are inconsistent:
- hasMaterial Metal



- Green bar: the system has decided the property is present because it has a probability higher than 70 %.
- Red bar: the system has decided the property is absent because it has a probability lower than 30 %.
- Orange bar: the system could not decide about the presence/absence of this property.

(b) An inconsistent case

Figure 5.8: Two examples of the explanation interface

values of the thresholds to better understand the link between the thresholds and the consistency of the system. This example lets the user generate their own contrastive explanations i.e. answer the question "What would change if the thresholds were modified?". Contrastive and counterfactual explanations have been successfully used by Vermeire et al. [176] and Pintelas et al. [55] to explain image classification. The goal of counterfactual explanations is to identify the minimum changes to make to an input in order to alter the output. This approach could be applied to OBIC not only to explain the image classification but also to explain the consistency. For instance, a user may wonder what changes in the assertions should be made to make consistent an inconsistent prediction e.g. "what modifications should be made on the assertions in the case illustrated in Figure 5.8b to make it consistent?". Proposing counterfactual explanations may increase the interactivity as the user chooses the goal (e.g. "what should change to make another class consistent?" or "what should change to make the current prediction consistent?"). It would also highlight shortcomings of the model or the ontology rendering the system more transparent. Unfortunately, there is no counterfactual explanations technique for ontologies. The contribution in Chapter 6 addresses this problem.

5.4 Conclusion

In this chapter, we introduced the design of a complete XIS according to the DARPA's schema [15]. This XIS is composed of an explainable model with a specific learning method as well as an explanation interface that presents the results of the model, shows relevant information and explains the predictions. The proposed explainable model, OBIC, utilizes an ontology to build multiple machine learning classifiers that can detect particular properties of an object in the data. The same ontology is then used as an error detection system by verifying the consistency of the predictions of these classifiers. The ability to detect and warn the user when a prediction is inconsistent is a step towards trustworthy and robust predictive systems. Indeed, we argue that this system is more robust than a single predictive model because it aggregates the results of concurrent statistical models using logic. The ability to warn the user when inconsistent predictions are detected as well as to provide the information to study the cause of the inconsistency increases transparency and thus trustworthiness. Furthermore, the design of OBIC is intended to minimize the amount of work required to implement it. The ontology at its core can be an existing ontology that only needs a few additional object properties and class definitions to be compatible with OBIC. Likewise, any model architecture can be used as the classifiers, allowing the reutilization of trained models instead of training new ones. Lastly, we explored the possible explanations that can be generated solely based on information extracted from OBIC. We designed a simple graphical explanation interface to present the results of OBIC (i.e. the class, detected properties and the overall consistency) as well as additional information that can help the user understand the prediction.

Although OBIC is intended to be easy to implement, there are several choices regarding the design of the ontology, the choice of thresholds and models. Depending on the available dataset, it may be difficult to identify observable properties that are characteristic of a class and that are observable on the majority of the dataset. In addition, we did not propose a methodology to determine the best choice of thresholds that has a direct impact on the performance of the error detection system. An evaluation of OBIC is conducted in Chapter 7 where we test our assumption that OBIC is easy to implement and evaluate the performance of the error detection system with regards to the choice of thresholds. Finally, the choice of model architecture was not discussed in the chapter but has an influence on the explainability of the entire system. We apply machine learning models to make the predictions despite their lack of interpretability and explainability. Since OBIC is model-agnostic, the choice of model architecture is made by the practitioner that implements this system. This choice is motivated by the task at hand. Some tasks will require high predictive performance and require the application of state-of-the-art models that are black-box. In this case, existing XAI methods can be used in addition to the explanations of OBIC to understand the behavior of each model. In other cases where explainability is preferred, interpretable

or self-explainable models can be employed.

The proposed explanation interface completes the XIS and provides information about the outcome of the system in order to increase transparency. It is currently not interactive and thus does not generate good explanations according to our definition of an explanation. Despite our best efforts to create an interface that is readable and understandable by most users, we believe that it requires major overhauls with experts in human-computer interactions to make it usable in actual applications. Furthermore, the proposed explanations of the classification is solely based on the detected properties. For instance, it is explained that a wooden chair was detected because the material wood and the color brown were detected. Yet, the *global classifier* does not communicate with the property models to make the prediction. Thus, the explanation is not faithful as it does not reflect the actual decision process. In addition, the interface only displays *raw explanations* [185] that need to be refined by explainability techniques in order to provide adequate explanations. We address this problem in Chapter 6 by designing a method that generates counterfactuals explanations for the error detection system.

The OBIC framework was developed with a bottom-up approach. The initial goal was to find a method that can explain the classification of images by using ontologies. We focused on this particular task because an image is the ideal data type to test and compare XAI methods. Indeed, most existing XAI methods are applicable to images, the resulting explanations are often visual and do not necessitate any language processing.

We claim that OBIC is an instance of a generic framework that is agnostic of the nature of the data. The underlying idea of OBIC is to use raw data to detect human concepts that are implicitly studied by humans to make a prediction. For instance, a human can tell that an image contains a wooden chair by analyzing concepts such as the material, the shape or the context of an object in an image. Inversely, a machine learning model functions pixel-wise and does not rely on such intermediate concepts to make the prediction. Hence, OBIC functions in two generic steps:

1. Exploit human knowledge to determine relevant intermediate concepts to observe and create an AI system to detect these concepts in raw data.

2. Apply logical reasoning on the detected concepts to make or confirm a decision, based on expert knowledge in the form of an ontology.

These steps are not data specific and can be applied to other tasks. The generalization of OBIC is further discussed in Chapter 8.

Chapter 6

Counterfactual explanations for ontologies

Contents

6.1 Literature review 99
6.1.1 Counterfactual explanations 100
6.1.2 Explaining ontologies 106
6.1.3 Similarity metrics for individuals 106
6.2 Counterfactual explanations for ontologies 109
6.3 The CEO method 114
6.3.1 Exploring the counterfactuals space 116
6.3.2 Computing metrics 121
6.4 Validation 122
6.4.1 Test cases and results 123
6.4.2 Analysis 124
6.5 Conclusion 126

The XIS discussed in Chapter 5 introduced the use of an ontology to detect inconsistencies in its predictions. The proposed explanations of these inconsistencies are minimal and scarce, which motivated the creation of another method to explain the inconsistencies. However, the literature on XAI methods is focused on explaining machine learning models, and explainability methods for ontologies are lacking. We noted in Chapter 3 that XAI methods are based on the functioning of interpretable models to represent the behavior of machine learning models. XAI methods that use surrogate models are not relevant to explain ontologies. Hence, we turned our attention to example-based explanations and especially counterfactuals as they have been identified as ideal candidates for explanations [33, 39].

In this chapter, we review the literature on counterfactual explanations and explanation techniques for ontologies. We also study similarity metrics for individuals as we will need to compute the distance between two individuals of the ontology for our contribution. Then, we introduce a novel method to generate Counterfactual Explanations for Ontologies (CEO). This method can be used to explain the outcome of the XIS described in Chapter 5.

6.1 Literature review

Counterfactual explanations are a well-known type of explanation in social sciences. Recently, the XAI community identified counterfactual explanations as an ideal way to explain AI systems and begun designing methods and metrics to generate these explanations for machine learning models. In this section, we first present counterfactual explanations:

1. The general definition and advantages of counterfactual explanations.
2. The existing methods to generate them for machine learning.
3. The metrics to evaluate these explanations.

Then, we discuss the solutions that explain ontologies to identify the shortcomings and needs of this problematic. Finally, we review the existing metrics to compare entities of an ontology as we will need to compare these entities for the design of counterfactual explanations for ontologies.

6.1.1 Counterfactual explanations

We discussed the notion of counterfactual reasoning in Section 3.1.1. It is defined as the process of identifying necessary causes of an event by hypothesizing what would happen to the event if some causes were different. In a psychology bulletin, Roese defined it as mental representations of alternatives to the past [192]. Likewise, Molnar described counterfactual reasoning as imagining a hypothetical reality that contradicts the observed facts, hence the name counterfactuals [54]. Counterfactual explanations answer the following question:

What should be changed to get Q instead of P ?

Wachter et al. [39] noted that there may be many different sets of actions to change the outcome. In order to avoid providing irrelevant explanations, the idea of seeking the *smallest* possible changes to modify the outcome is added to the definition of a counterfactual explanation [39, 61, 193]. The definition of counterfactual explanation is consensual in the literature. Definition 6.1.1 uses the formulation given in [193] that is adapted to XAI. Counterfactuals belong to the family of *example-based* explanations since they provide examples to explain a decision. In the following, the terms *counterfactual explanation(s)* and *counterfactual(s)* are used interchangeably.

Definition 6.1.1 (Counterfactual explanation)**.** A counterfactual explanation for a prediction highlights the smallest change to the feature values that modifies the prediction to a predefined output.

We introduce Example 6.1 to illustrate counterfactuals. This example of loan approval is commonly used in the literature, Verma et al. [60] thoroughly described it hence we use their formulation.

Example 6.1 Suppose a bank's customer seeks a loan. The loan approval system uses a classifier, which studies the customer's file. This file contains information on the customer's identity and financial situation. The feature vector is $(Income, CreditScore, Education, Age)$. When the customer is denied the loan by this system, they may ask for some explanations about this decision: "Why was the loan denied ?" and "What can I do differently so that the loan will be approved in the future ?". The first question can be answered using current explainability methods. A probable answer to that first question might be "Your $Income$ is too low". The second question clearly requires a counterfactual explanation: what are the smallest changes that the customer can do in order to change the outcome i.e. get the loan. A possible counterfactual may be: "Your $Income$ should be of 40K$ instead of 30K$". Another could be: "Your $Education$ should be master's degree instead of a bachelor's and your $Income$ should increase by 4K$." These formulations allow the customer to choose between different paths in order to get the loan and understand which variables are the most important for the model.

The term *contrastive* explanation is also used by scholars to refer to a similar process. Lipton [194] described a contrastive explanation as the answer to the question "Why P rather than Q?" where P is the event that happened and Q is a different event. On the contrary, counterfactual explanations answer the question "What should be changed to get Q instead of P ?". Stepin et al. [61] discussed the distinction between contrastive and counterfactual explanations. They stated that contrastive explanations point to the difference between the actual event (P) and a hypothetical one (Q) while counterfactual explanations specify necessary minimal changes in the input so that a contrastive output is obtained. Guidotti [193] argued that in the context of XAI, there is little difference between counterfactual and contrastive explanations. In both cases, the aim is to find what would have changed the decision. Nevertheless, Miller [195] nuanced this observation and warned that although counterfactuals contribute to contrastive explanations, they are not the same and using the terms interchangeably may result in terminology issues. We consider that counterfactuals participate in building contrastive explanations. Consequently, counterfactuals and contrastive explanations share the same qualities that will be discussed below.

In Section 3.1.1, we have briefly discussed the gain of interest of the XAI community for contrastive and counterfactual explanations. This rise in popularity is due to the strong use of counterfactual explanations by humans to explain [33, 60, 61, 196]. Byrne [197] linked counterfactuals to the XAI problem and further discussed the uses of counterfactuals in order to maximize their effectiveness. Two user-studies were conducted [198, 199] to assess and compare the effectiveness of four explanation styles. These explanations are designed to enable users to judge the fairness of a model. They found that counterfactual explanations (named sensitivity based explanations in these articles) are convincing, easy to process and particularly effective to judge the fairness of a model compared to the case-based explanations. In addition to these findings, scholars argued that the generation of counterfactual explanations is technically feasible and GDPR-compliant[1] [39, 196]. The combination of all these qualities attributed to counterfactual explanations led to a recent explosion in the number of counterfactual explanation methods.

Despite these promising qualities coming from theories and intuitions, these methods have not been sufficiently validated with user-studies. Keane et al. [62] discussed this issue and observed that 25% of counterfactual methods conducted a user-study to validate the relevance of their approach. Moreover, only 7% of the papers they reviewed contained a comparison of several methods. Finally, an overwhelming majority of counterfactual explanation methods focus on explaining machine learning models. Although some methods are model-agnostic, they require a numerical input vector to function. In the following, we review the existing literature about counterfactuals for machine learning.

Counterfactual explanations for machine learning

Wachter et al. [39] proposed a machine learning oriented definition for counterfactuals: “Score p was returned because variables V had values $(v_1, v_2, \ldots)$ associated with them. If V instead had values $\left(v'_1, v'_2, \ldots\right)$, and all other variables had remained constant, score p' would have been returned”. Definition 6.1.2 formalizes counterfactuals specifically for machine learning models. In the remainder of this chapter, we use the notations introduced in this definition.

Definition 6.1.2 (Counterfactual for machine learning)**.** Given f a classifier and $x \in \mathcal{X}$ an input vector such that $f(x) = y$ where $y \in \mathcal{Y}$ is the predicted class. A counterfactual is a vector $\widehat{x} \in \mathcal{X}$ that follows these two constraints:

$$f(\widehat{x}) = \widehat{y} \tag{6.1}$$

where $\widehat{y} \in \mathcal{Y}$ is the *foil* class i.e. $\widehat{y} \neq y$.

$$\widehat{x} = \underset{x' \in \mathcal{X}}{\arg\min}\, d\left(x, x'\right) \tag{6.2}$$

[1] Wachter et al. [39] highlighted three requirements of an explanation in GDPR: understand, contest and alter decisions. Counterfactual explanations achieve these three requirements.

where d is a proximity metric, that measures the difference between the original input and a counterfactual.

We note that a variant of this definition exists where Equation (6.1) simply ensures that the counterfactual leads to a different outcome than the original input i.e. $f(\widehat{x}) \neq y$. For instance, Equation (6.1) ensures that the counterfactual leads to the loan being approved e.g. $\widehat{y} =$ `Approved` while the other definition is less restrictive and requires that the counterfactual leads to any class that is not the original one i.e. `Denied`. We argue that this variant is equivalent to Definition 6.1.2 in the case of binary classifiers. Still, contrastive explanations require a *foil* class i.e. a desired class [195]. In practice, counterfactuals methods use Definition 6.1.2 and ask for a desired class [39, 200, 201].

Counterfactuals in machine learning enable the user to identify the decision boundary between the original class y and the foil class $\widehat{y}$ [59, 60]. However, users may want to know the changes to make to get a particular outcome. The minimality constraint (i.e. Equation (6.2)) ensures that the closest solution is given to respect Definition 6.1.1 which requires the counterfactual to highlight the smallest changes. The metrics to measure the difference between the original input and the counterfactual are debated in the literature along with other desired properties that counterfactual explanations should have to maximize their effectiveness.

Several reviews of the literature listed the most widely used and shared desirable properties of a counterfactual [45, 59, 60].

Validity A counterfactual $\widehat{x}$ is valid iff it actually changes the classification outcome to the desired one i.e. it verifies Equation (6.1) [39, 60, 193].

Sparsity It measures the number of features that are different between the counterfactual and the original input. This property is promoted in several papers where authors advocate for short explanations, i.e. sparse counterfactuals [39, 59, 60]. This idea stems from theories on human working memory limits or other theories on human limitations and as a result, it is reported that there are "ideal" levels of sparsity. Yet, Keane et al. [62] argued that this property is based on intuition and the mentioned theories may not be applicable to this context. This debate relates to the notions of size or compactness discussed in Section 3.1.3. Guidotti [193] discussed the related property of minimality to ensure that the counterfactual is as sparse as possible. A counterfactual $\widehat{x}$ is minimal iff $\nexists \widehat{x}'$ s.t. $sparsity(x, \widehat{x}') < sparsity(x, \widehat{x})$.

Proximity Proximity is the metric used to measure the difference between the original input and a counterfactual. This metric is necessary to generate counterfactuals and corresponds to the function d in Equation (6.2).

Proximity is usually a feature-wise distance [61]. Verma et al. [60] mentioned the L_1 or L_2 distances as potential candidates for this metric. Wachter et al. [39] said that the choice of this metric is subject and task specific. They chose to define proximity as the mean of the feature-wise L_1 distances normalized by the median absolute deviation (MAD) of each feature. Mothilal et al. [59] took the same approach to measure the difference between continuous features. However, the input feature space is often heterogeneous [201] containing continuous and categorical features. Example 6.1 illustrates this problem, the $Income$ feature is continuous while $Education$ is categorical.

Finding adequate metrics to evaluate the difference between categorical features is challenging. Indeed, not all counterfactual methods are able to handle them [193]. Mothilal et al. [59] mentioned that a metric for categorical features should represent the *difficulty* of changing a particular feature. They fell back on the sparsity metric to determine the difference between categorical features then they calculate the mean sparsity i.e. $\frac{1}{n}\sum_i 1_{x_i \neq \widehat{x}_i}$ where n is the number of categorical features. Karimi et al. [201] distinguished numerical, categorical and ordinal features. For instance, hair color is categorical, education level is ordinal and income is numerical.

- For numerical and ordinal values, the feature-wise distance is $\|x_i - \widehat{x}_i\|_1 / R_j$ where R_j is the range of the feature x_j.
- For categorical features, they used the same metric described previously i.e. $1_{x_i \neq \widehat{x}_i}$. The distances are represented by the vector δ where δ_i is the computed distance between x_i and $\widehat{x}_i$.
- Finally, they used a linear combination of L_p norms to get the proximity: $d(x, \widehat{x}) = \alpha \|\delta\|_0 + \beta \|\delta\|_1 + \gamma \|\delta\|_\infty$.

This choice is motivated by the properties of each norm, $\|.\|_0$ restricts the number of features that changes thus minimizes sparsity, $\|.\|_1$ restricts the average change and $\|.\|_\infty$ restricts the maximum change across features. We observe that the sparsity property is usually included in the proximity metric with the L_1 norm.

Plausibility Also known as *feasibility*. It is a measure of whether a counterfactual is realistic or makes sense in the real world [193, 196, 202]. Guidotti [193] argued that plausibility helps in increasing trust towards the explanation. Indeed, a counterfactual that proposes unrealistic changes cannot be trusted. For instance, changing the $Income$ from 30 000$ to 1M$ is not plausible. Likewise, some changes may not be possible such as decreasing one's age. There are several approaches to increase plausibility used in the literature. The most prominent approach to create a plausible counterfactual is to ensure that it follows the data distribution [60, 193, 201] or remains within the range of a given feature [200]. Using the Example 6.1, a customer of 18 years old with a high school diploma is denied a loan. A counterfactual requiring this customer to get a doctorate's degree and be 20 years old is unrealistic and surely falls outside the data distribution. Hence, advocates of this approach consider that the data distribution is representative of the reality. Keane and Smyth [196] proposed a case-based method to find counterfactuals. Therefore the proposed counterfactuals are instances of the original dataset making them plausible. However, this method is confronted to the lack of good counterfactuals that are "naturally" available in datasets. Actionability and causality described below are other properties to increase the plausibility of a counterfactual.

Actionability A counterfactual is actionable when it only modifies actionable features [59, 61, 193, 201]. A feature is actionable if it is fair and feasible to mutate it. A typical non-actionable feature is the age because it is an immutable feature of an individual. The actionability of a feature is determined by the user or domain experts.

Causality Counterfactuals should respect causal relationships to be plausible [60, 193]. It is yet another method to reinforce the plausibility of a counterfactual. Features in the input space are causally related. For instance, there are causal relationships between $Income$, Age and $Education$. Indeed, the level of education is known to have an effect on the income and similarly, age also has an effect on education level as well as income since the income usually increases with the experience.

Several methods discussed the use of user-defined constraints that determine implausible changes in some features based on known causal relationships [59, 193, 201, 202, 203]. A user-friendly language to declare the known causal relationships is sometimes provided to facilitate the declaration of these relationships [59, 202]. For instance, the DiCe method [59] allows the user to define ranges for specific features and decide the relation between pairs of features. These relations dictate the possible evolution of one feature in regards to another e.g. when $Education$ increases in a counterfactual, then Age should also increase.

Diversity It measures the difference between counterfactuals. Most proposed algorithms return a single counterfactual for a given input [60]. Nevertheless, researchers have been working on generating a set of counterfactuals to explain one input since it is argued that providing several counterfactuals increases the comprehension of the observed model as well as the number of possible paths to modify the outcome [39, 59, 202]. The measure of diversity

ensures that there is as little overlap as possible between each counterfactual. This measure is analogous to the proximity metric but measures the distance between two counterfactuals instead of a counterfactual and the original input. Diversity should be maximized to gain as much information as possible.

Mothilal et al. [59] used determinantal point processes to capture the diversity of a set of counterfactuals. It is computed based on the determinant of the kernel matrix K such that $K_{i,j} = 1/\left(1 + d\left(\widehat{x}^i, \widehat{x}^j\right)\right)$ where $\widehat{x}^i$ and $\widehat{x}^j$ are two distinct counterfactuals and d is a metric, usually similar to the proximity metric.

In recent years, many methods to generate counterfactuals for machine learning have been designed. They all attempt to solve the optimization problem formulated in Definition 6.1.2. Scholars have applied different strategies to find solutions to this problem that also satisfy the other desired properties discussed above. According to Guidotti [193], two major strategies are commonly applied: the resolution of the problem with optimization algorithms and the resolution via a heuristic search. Heuristic search is more efficient than the other approach but returns suboptimal solutions. Both strategies attempt to minimize a loss or cost function. The difference between each method is the definition of this cost function.

Wachter et al. [39] were among the first to propose a method to generate counterfactuals for machine learning. Consequently, the loss function is solely based on the definition.

$$\arg\min_{\widehat{x}} \lambda \left(f\left(\widehat{x}\right) - \widehat{y}\right)^2 + d\left(x, \widehat{x}\right) \tag{6.3}$$

Equation (6.3) is minimized to find a candidate counterfactual. The value λ balances the contribution of the first term against the second term [193]. A low value of λ favors the proximity while a high value enforces that the prediction is equal to the expected outcome. The authors use an optimizer to find the solution of this problem. Equation (6.3) is used by other methods as a foundation for their cost function. Verma et al. [60] showed that terms are added to this foundational cost function in order to include additional constraints such as plausibility, sparsity or diversity.

Mothilal et al. presented DiCE (Diverse Counterfactual Explanations) [59], a method that also uses an optimizer to find counterfactuals. This method is capable of generating several counterfactuals in one run. Hence the optimization problem must be formulated differently to generate k counterfactuals instead of one and take into account the diversity problem specific to the generation of multiple counterfactuals.

$$\arg\min_{\widehat{x}_1,\ldots,\widehat{x}_k} \frac{1}{k}\sum_{i=1}^{k} \text{yloss}\left(f\left(\widehat{x}_i\right), y\right) + \frac{\lambda_1}{k}\sum_{i=1}^{k} d\left(\widehat{x}_i, x\right) - \lambda_2 \text{diversity}\left(\widehat{x}_1, \ldots, \widehat{x}_k\right) \tag{6.4}$$

The optimization problem is formulated in Equation (6.4) and finds the list of generated counterfactuals for input x, $yloss$ enforces the change of outcome of each counterfactual and $diversity$ calculates the diversity based on the determinantal point processes discussed above. The $diversity$ function is a regularization term that penalizes non diverse solutions. They do not include sparsity and plausibility constraints in the optimization problem but instead apply a filter on the resulting counterfactuals. They greedily restore the value of continuous features until the predicted class changes to encourage sparsity. Concerning plausibility, they discuss the possibility of adding user-constraints that indicate actionable features and causal relations between features. These constraints are also checked after solving the optimization problem. Counterfactuals that do not satisfy the user-constraints are removed from the presented solution. In the conducted experiments, they estimate that a third of the counterfactuals are not plausible and should be removed based on a set of causal relationships.

Sharma et al. introduced CERTIFAI [204], to generate counterfactuals and assess the robustness and fairness of a classifier. It uses a genetic algorithm to solve Equation (6.3). The set of candidate individuals is defined and explored iteratively to find the individuals that minimize the cost function. This algorithm supports the addition of constraints to ensure plausibility e.g. the values

of income should remain in the range [10 000, 100 000] or the nationality should be immutable. CERTIFAI computes a robustness and fairness score for a given classifier by analyzing the generated population of counterfactuals. For robustness, the expected distance between the input instances and the corresponding counterfactuals is calculated. The fairness score is calculated by generating counterfactuals with different values of a protected (non-actionable) feature and checking that these counterfactuals are not easier to achieve.

Schleich et al. designed GeCo [202] which also uses a genetic algorithm to find counterfactuals. They defined the initial space of candidates to be explored based on two components: a database of entities that contain real world examples (e.g. historical, training or test data) and a set of plausibility constraints via their novel language called PLAF. The database is used to extract the plausible range of each feature automatically and allow the data analyst to create groups of features that are causally linked. The authors provide the examples of the zip code and the city, these two features are functionally dependent and should be grouped together. Likewise, education level and income are correlated and may be grouped together. Then, the PLAF language enables the data analyst to precise the nature of the relationship between features. This language functions with the definition of predicates of the form $e_1 - \text{op} - e_2$ where e_1 and e_2 are features of the original input or the counterfactual and op is a mathematical operator in the set $\{=, \neq, \leq, <, \geq, >\}$. For instance, the rule `x_cf.nationality = x.nationality` imposes that the nationality is not changed, rendering nationality a non-actionable feature. The rule `IF x_cf.education > x.education THEN x_cf.age > x.age + 4` implies that if the education level in the counterfactual is greater than the one in the original input, then the age must also be increased by at least 4 years. We note that this language resembles a rule-based system where the rules are declared by an expert based on their knowledge. This solution avoids the drawback of DiCE [59] that checked plausibility and removed implausible solutions after the generation process.

Counterfactual methods are usually evaluated on objective metrics and there is a lack of user studies that validate these metrics [61, 62]. The evaluation metrics are mostly similar to the desired properties of counterfactuals i.e. proximity, sparsity, diversity, validity and plausibility. Guidotti [193] proposed an extended set of evaluation metrics and benchmarked all the methods reviewed in the same paper. Still, scholars mentioned the difficulty to produce fair comparisons between methods since neither the metrics nor the functioning of the methods are standardized. Indeed, Schleich et al. [202] could not apply their proposed proximity metric to some methods since these methods do not support this form of metric. Moreover, methods that generate a single counterfactual do not consider diversity in their design which may unfairly penalize them in benchmarks.

The state of the art on counterfactuals for machine learning reveals several limitations. Although there is a consensus on the desired properties and the related terminology, how to achieve this property is still unclear. The most evident example is the computation of a proximity metric for categorical features. As Mothilal et al. [59] pointed out, the proximity metric applied feature-wise should represent the difficulty of changing the feature from the original value to a new one. Indeed, counterfactuals provide possible actions on one's current situation to modify the outcome of a decision. However, the proposed proximity metrics are not able to measure this difficulty for categorical features. Similarly, evaluation metrics are not agreed upon and as a result, the comparisons between methods are often unfair. Moreover, few user-studies have been conducted which may be the best way to validate and compare different methods. Finally, the notion of plausibility seems crucial to generate good counterfactuals as it ensures that the changes one has to make to modify the outcome are coherent and feasible. Yet, there is no fully automated method that extracts plausibility constraints and enforces them. Some methods automatically check that a counterfactual is within the data distribution although the relevance of this technique is debated since it assumes that outliers are not plausible. Other methods look for real counterfactual instances which are guaranteed to be plausible but the amount of candidate counterfactuals is usually very limited. Scholars introduced languages to let the user or practitioner input their own set of constraints e.g. causal relations or actionable features. We observe that these languages mimic rule-based systems that extracted rules from expert knowledge. To our knowledge, no method uses

a knowledge-based system to automatically extract the rules. Additionally, we argue that plausibility is not always desirable as it may prevent the detection of unfair AI systems. As a matter of fact, implausible counterfactuals may reveal the unfairness of a model. Indeed, the application of counterfactuals on the COMPAS dataset by Mothilal et al. [59] showed that changing only a protected feature led to a change of outcome. Likewise, CERTIFAI relies on implausible counterfactuals to compute its fairness score [204].

6.1.2 Explaining ontologies

Explanations in OWL ontologies are necessary to help a designer or a user understand entailments, debug and repair an ontology [205]. Since OWL ontologies are based on description logics, it is possible to extract some explanations of entailments by using a reasoner. Methods to generate explanations are divided into two types: black-box and glass-box methods [206]. According to [207], *glass-box* methods introduce significant modifications to description logic reasoners with the goal to use available internal information for a fast computation of diagnoses. *Black-box* methods use a reasoner as an oracle to check if some set of axioms is consistent. We note that this terminology is specific to logical reasoners and will not be used outside this section to avoid any ambiguity.

The simplest type of explanations that can be extracted from the reasoner is logical proofs. They display each step of the reasoning process that resulted in a specific entailment. The main issue with such explanations is that they become difficult to understand when they get very large [208]. As a response to this problem, another form of explanation called justifications are introduced [209]. They are also called MUPS for Minimal Unsatisfiability-Preserving sub-TBoxes. They consist in finding the smallest sets of axioms necessary for a given entailment to hold. However, as Alrabbaa et al. [210] mentioned, justifications can still be very large and thus suffer from the same issue as proofs.

In order to overcome these issues, interactive debugging tools have been proposed. OntoDebug [207] implements this idea. Its goal is to ask the user for additional knowledge that can reduce the length of proofs and justifications. However, Coetzer and Britz [211] have shown that the debugging approach of OntoDebug can lead to unintuitive results. Despite all the efforts in the development of debugging tools, ontology authors still struggle to debug and repair their ontologies [212].

The explainability of OWL ontologies is confronted with the same issues as XAI, with a similar goal. Both seek to provide understandable explanations of decisions made by an algorithm. In the XAI field, such algorithms are machine learning algorithms whereas for OWL ontologies, they are reasoners. Interestingly, there is some shared terminology, e.g. glass-box and black-box, that also share the same notions. Finally, as Lecue [31] advocated, ontologies could benefit from the advances in XAI in the same manner as the XAI field benefits from the knowledge-representation and reasoning domain. Indeed, in the majority of the reviewed literature on OWL explanations, the explanations are made only for domain experts and ontology authors. Providing explanations to laypersons could help ontologies gain popularity and be used as trustworthy decision-support systems.

6.1.3 Similarity metrics for individuals

The generation of counterfactual explanations for ontologies requires to compute proximity among entities of an ontology. Many metrics that measure distances between ontologies and between concepts of an ontology have been developed over the years. First and foremost, we define distance, similarity and dissimilarity functions to understand the conditions that each metric must respect, based on the definitions given in [213].

Definition 6.1.3 (Dissimilarity)**.** Given two entities in the same space Ω, a dissimilarity $\delta : \Omega \times \Omega \rightarrow \mathbb{R}$ is a function from a pair of objects to a real number such that:

$$\forall x, y \in \Omega, \delta\left(x, y\right) \geq 0 \tag{6.5}$$

$$\forall x \in \Omega, \delta(x,x) = 0 \tag{6.6}$$

$$\forall x, y \in \Omega, \delta(x,y) = \delta(y,x) \tag{6.7}$$

Definition 6.1.4 (Similarity)**.** A similarity $\sigma : \Omega \times \Omega \rightarrow \mathbb{R}$ is a function expressing the similarity between two objects that is positive and symmetric i.e. satisfies Equations (6.5) and (6.7). In addition, a similarity function respects the maximality constraint defined in Equation (6.8). The similarity of two identical objects return the upper bound of this function.

$$\forall x, y, z \in \Omega, \sigma(x,x) \geq \sigma(y,z) \tag{6.8}$$

Definition 6.1.5 (Distance)**.** A distance $d : \Omega \times \Omega \rightarrow \mathbb{R}$ is a dissimilarity function satisfying definiteness and the triangular inequality, defined respectively in Equations (6.9) and (6.10).

$$\forall x, y \in \Omega, d(x,y) = 0 \iff x = y \tag{6.9}$$

$$\forall x, y, z \in \Omega, d(x,y) + d(y,z) \geq d(x,z) \tag{6.10}$$

A dissimilarity function defined in Definition 6.1.3, must follow three conditions: positiveness (Equation (6.5)), minimality (Equation (6.6)) and symmetry (Equation (6.7)). A distance function defined in Definition 6.1.5 has more constraints than a dissimilarity function but works in the same way i.e. two identical objects return a dissimilarity of 0. The distance function is more definite as a distance of 0 ensures that the objects are the same which is not the case for dissimilarity. Inversely, a similarity function has the same definition than dissimilarity with the difference that two identical objects return the upper-bound of the function. Euzenat et al. [213] argued that there are many reasons why an ontology measure may not be a distance and give the example of two semantically equivalent concepts. It is expected that equivalent concepts return a distance of 0 even if they are not the same, motivating the removal of the definiteness condition. As a result, most semantic metrics are similarity or dissimilarity functions. These measures are usually normalized (ranging from 0 to 1), meaning that it is easy to transform a normalized dissimilarity intro a normalized similarity by using its complement to 1 [213]. Hence, we use the term similarity to refer to both dissimilarity or similarity functions.

Two strategies to measure the distance or similarity of ontology entities are identified by scholars: syntactic and semantic [214] (or intensional and extensional [215]). Syntactic or intensional approaches exploit the structure of the ontology e.g. the concepts definitions or the relationships between concept. Semantic or extensional approaches use the set of instances to measure probability distributions or concept co-occurrences. An unbiased population of instances is assumed which may not be applicable to all ontologies. Indeed some ontologies have little to no individuals which prevents the use of such approaches.

Fernández-Chamizo et al. [216] introduced syntactic similarity measures for individuals and concepts. Concerning individuals, the similarity is computed as the sum of two factors. The first factor is the similarity between the concepts of which the individuals are instances. The second factor is the similarity among the relations of the individuals i.e. the assertions where the individual is the `subject`. The similarity between individuals o_1 and o_2 is defined in Equation (6.11).

$$sim(o_1,o_2) = \begin{cases} sim_c(t(o_1),t(o_2)) & \text{if } \forall r \in R, o_1.r = \emptyset \text{ or } o_2.r = \emptyset. \\ \frac{1}{2}\left(sim_c(t(o_1),t(o_2)) + \frac{\sum_{r \in R} sim_r(o_1.r,o_2.r)}{|\{r \in R | o_1.r \cup o_2.r\}|}\right) & \text{otherwise.} \end{cases} \tag{6.11}$$

where $o.r$ corresponds to the set of individuals that are objects of the assertions $\texttt{r}(\texttt{o, object})$, R is the set of all predicates, $t(o)$ returns the concepts of which entity o is an instance. The functions sim_c and sim_r correspond to the similarity measure between concepts and sets of individuals respectively. To calculate the similarity between concepts, the authors considered that every concept in the ontology is an attribute. The attributes of a concept are itself and its parent concepts. These attributes are represented in a vector v such that v_i corresponds to the i-th concept. The

value of v_i is equal to 1 if concept i is an attribute possessed by the entity, 0 otherwise. The function sim_c is the cosine similarity between the vectors of each entity. Finally, sim_r computes the similarity between relations of the same nature (i.e. with the same predicate) of two individuals. The sets of individuals related to o_1 and o_2 with the predicate r are compared with the function sim_r. This function recursively calls the function given in Equation (6.11) on every combination of individuals.

Hu et al. [215] proposed an approach similar to the vectorization described in [216] when measuring the similarity of concepts. They define primitive concepts as concepts that are only defined by names. A concept C can be unfolded into a set of primitive concepts. A weight is attributed for each primitive concept, based on the number of occurrences of the primitive concepts in the definition of C. A vector containing the weight for each primitive concepts that appears in the definition of C is built and named the signature vector of C. The similarity of two concepts is now the similarity of their respective signature vectors.

Janowicz [217] presented SIM-DL, another syntactic similarity for description logics. This similarity measures the overlap between the descriptions of two concepts. Concepts C and D are defined as $C = C_1 \cup \cdots \cup C_n$ and $D = D_1 \cup \cdots \cup D_m$. The similarity between these concepts is given as $sim_u(C,D) = \sum_{(C_i,D_j)\in SI} w_{i,j} \times sim_i\left(C_i, D_j\right)$ where sim_i computes the similarity between the concepts C_i and D_j that are represented in another form than C and D. The pairs C_i and D_j are selected by iterating over each C_i and matching it with the most similar concept D_j, resulting in the set SI of selected pairs. The weights $w_{i,j}$ act as adjustable factors to reflect the relative importance of each pair, the sum of all $w_{i,j}$ adds up to 1. In turn, sim_i recursively calls other similarity measures between the primitive concepts, existential, value and number restrictions/quantification of concepts C_i and D_j. This method is specifically designed for $\mathcal{ALCNR}$ descriptions (see Section 3.2.2).

Euzenat et al. [213] described the OLA similarity which first encodes an ontology as a labeled graph. The similarity between two nodes of this graph depends on the similarity of the terms (labels, names...), the similarity of the neighbors and the similarity of other local descriptive features. Ghosh et al. [218] also discussed the use of graph-based semantic measures to determine similarities of entities in an ontology. Moreover, we argue that the classes of an ontology can also be represented as a hierarchical tree or a taxonomy which enables the use of other types of similarity metrics. Ontañón [214] presented several similarity functions for structured data, including hierarchies, taxonomies and graph-based representations.

A hierarchy is a partially ordered set where each element has at most one parent [214]. Two distance functions that are commonly used to compare elements of a hierarchy are the edge-counting [219] and information content [220] functions. Rada et al. [219] introduced the edge-counting distance function, where the hierarchy is seen as a tree. The elements are nodes of the tree and the parent relationships are edges linking elements together. The distance is the number of edges that need to be traversed to reach element b starting from element a. Resnik [220] improved this edge-counting distance. Each element of the hierarchy is a concept that can be instantiated. This similarity measure gives a weight to each concept that reflects the probability of encountering an instance of that concept. This probability distribution is called the information content of a node. The similarity measure proposed by Resnik looks for the least common subsumer (LCS) i.e. the common parent of two elements, and uses the information content of this LCS to measure the similarity between two elements. This method is better than edge-counting since it reflects the importance of each concept rather than attributing the same value to each concept. However, computing the information content of each concept requires a set of instances that may not be available.

Finally, a similarity measure that may take advantage of the graph representation of an ontology is the Graph Edit Distance (GED) [221]. Given a set of elementary graph operations (e.g. deletion, insertion and substitution of vertices or edges), the graph edit distance between two graphs

g_1 and g_2 is defined in Equation (6.12):

$$GED\left(g_1, g_2\right) = min_{e_i \in \mathcal{P}(g_1, g_2)} \sum_{i=1}^{n} c\left(e_i\right) \tag{6.12}$$

where $\mathcal{P}\left(g_1, g_2\right)$ is the set of elementary operations to transform g_1 into g_2 and $c\left(e_i\right)$ is the cost of the graph operation e_i. The cost for each operation needs to be defined based on the task. The GED gives a good framework to compare two graphs while giving freedom on the choice of cost function for each operation.

We have discussed several metrics to compute the similarity between ontological entities. We notice some common ideas in the functioning of these metrics. Some unfold concepts and recursively call different similarity measure depending on the nature of the entity (mostly concepts and individuals). A drawback of these methods is that unfolding recursively may lead to infinite loops when concepts are cyclically defined. Others identify primitive concepts to vectorize individuals or other concepts. The class of extensional measures require a set of unbiased instances to be effective which may not exist for all ontologies. Finally, similarity measures created for graphs or taxonomies can also be applied with minimal effort to adapt them for ontologies.

6.2 Counterfactual explanations for ontologies

The literature review revealed that methods to explain ontologies are intended for ontology experts for debugging purposes. To our knowledge, there is no method designed to explain ontologies to laypersons. The desire to generate explanations for our XIS as well as Lecue's observations [31] that ontologies could benefit from XAI motivated us to produce an XAI method dedicated to explaining ontologies. We discussed the many advantages of counterfactual explanations in Section 6.1.1 that led to a rise of popularity of these explanations in XAI. Consequently, a large number of XAI methods that generate counterfactuals have been proposed in the recent years. These methods are designed to explain machine learning models and none are designed to explain symbolic AI algorithms since they are considered interpretable. Our review of the literature also exposed several problems in the generation of counterfactuals for machine learning. Namely, the computation of an adequate proximity metric for categorical features, the lack of human subject evaluations, the difficulty to determine the conditions for plausibility and generate plausible counterfactuals. The problem of defining a proximity metric for categorical features is mostly overlooked in the literature. Current solutions fail at representing the difficulty to modify such feature from one value to the other because of the subjective and domain-specific nature of this problem. Human subject evaluations to assess the quality and relevance of the counterfactual explanations are lacking and prevent the XAI community from validating their methods and objective evaluation metrics. Likewise, the definition and generation of plausible counterfactuals is usually discussed but never evaluated with user studies. Some methods propose solutions to let the user define their plausibility constraints but this solution is never submitted to actual users for evaluation. In addition, scholars also discussed the problems of stability and robustness. Guidotti [193] noted that current methods are unable to handle missing attributes since they rely on a notion of distance between instances, thus all instances must have the same set of attributes.

We argue that ontologies may be an ideal tool to handle the computation of proximity between categorical features and determine the plausibility of a counterfactual. Scholars advocated or used methods based on expert knowledge to solve these issues (e.g. the PLAF language [202]). Ontologies can be used to check plausibility via checking the consistency of a counterfactual. If a counterfactual is not consistent, then it contradicts existing rules dictated by a domain-expert rendering it implausible. This may not always be true when the design of the ontology is faulty. In this case, the method serves as a debugging tool to improve the ontology and fulfills the "explain to improve" goal of XAI. Concerning stability and robustness issues, ontologies and logical reasoners are not susceptible to such issues as the inferences are deductive and deterministic i.e. a same input always leads to the same output and perturbations are not a concern because of the

discrete nature of entities in an ontology. OWL ontologies use the open-world assumption (see Section 3.2) meaning that it is capable of handling missing attributes contrary to current counterfactuals methods.

These observations motivate the creation of Counterfactual Explanations for Ontologies (CEO). The main contribution is an explanation method for ontologies that is comprehensible by everyone. A consequent contribution is the creation of the premises for a formalization that connects OWL ontologies to machine learning. This formalization could help adapt XAI methods from machine learning to ontologies and inversely, facilitate the use of ontologies for machine learning specific XAI methods e.g. replace the PLAF language from [202] with an ontology. Our approach is to develop the CEO method based on the current work in machine learning. The remainder of this chapter is organized as follows:

1. Define the counterfactuals problem for OWL ontologies i.e. how to represent the input and the resulting counterfactuals as well as how to formulate the desired properties to be compatible with this representation

2. Design the algorithm to generate counterfactuals.

3. Identify and find solutions to the problem of computing proximity and sparsity for ontologies.

4. Validate CEO.

Counterfactual explanations for machine learning classifiers are generated by searching for a vector that is similar to the original input but leads to a different output when processed by the model. This new vector should also satisfy the properties listed in Section 6.1.1 as much as possible. The concepts of input vector, model or prediction are specific to machine learning algorithms. Hence, similar concepts must be defined for ontologies in order to apply the same process. Namely, what are the input, output and model of a method that generates counterfactuals for ontologies ? Moreover, how to define and compute the desired properties of a counterfactual based on these new notions ? In the following section, we study a basic example in order to get a grasp of what a counterfactual for an ontology is and how it compares to machine learning. Then we propose a definition of a counterfactual explanation in the context of an ontology.

In Example 6.1, we presented the example of a customer that seeks a loan. This customer is represented by a set of features that are their annual income, credit score, education and age. In the case of machine learning, these features were given as input in the form of a vector. We argue that each customer would correspond to different individuals that populate the A-Box. For instance, let one customer named *Alice* with the following file: their annual income is 30 000$, their credit score is 500, they have a master's degree and are 27 years old. The definition of the individual *Alice* in the ontology's A-Box is the following set of assertions:

- $(Alice - \texttt{:hasIncome} - 30\,000\$)$
- $(Alice - \texttt{:hasCreditScore} - 500)$
- $(Alice - \texttt{:hasEducation} - master)$
- $(Alice - \texttt{:hasAge} - 27)$

The T-Box of an ontology can be compared to the parameters of a machine learning model, it contains knowledge under different forms e.g. the rules to approve or deny a loan. Particularly, it contains the definition of classes that allow a logical reasoner to make inferences. For example, a person is of class `Denied` when their annual income is less than 35 000$. The class `Approved` is defined as any individual that is not of class `Denied`. The logical reasoner will therefore attribute the class `Denied` to *Alice*. Thus, classes of an individual can be compared to the output of a classifier.

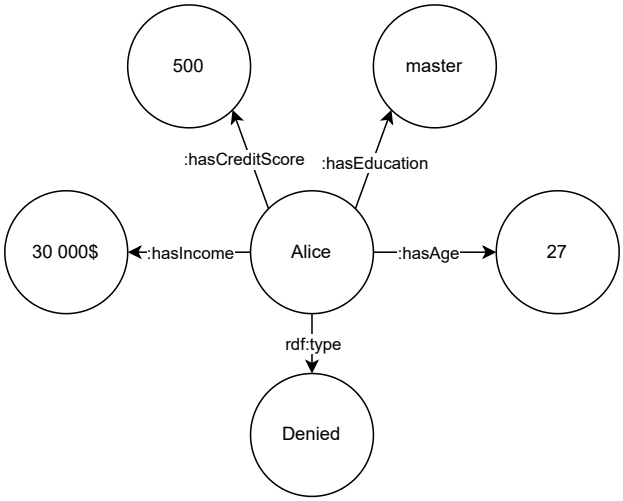


Figure 6.1: An IKG representing the customer *Alice*, classified as `Denied`

The goal of a counterfactual is to determine the minimum changes to get a different outcome. In this case, the different outcome is changing the class attributed to *Alice* from `Denied` to `Approved`. Based on the definition of the class `Approved`, we expect the counterfactual to change the assertion $\big(Alice - \texttt{:hasIncome} - 30000\$\big)$ to $\big(Alice - \texttt{:hasIncome} - 35000\$\big)$. From these intuitions, we gather that the input of the CEO method is an individual with its set of assertions. A logical reasoner exploits the ontology to act as a classifier and the output is another individual with a modified set of assertions. However there are some caveats specific to OWL ontologies such as the open world assumption that require adapted solutions.

Based on this discussion, we define the structure of the input and generated counterfactuals. We have observed that the input vector corresponds to an individual and its set of assertions. This set of assertions can be represented as a graph that we call an Individual's Knowledge Graph (IKG). Definition 6.2.1 defines this notion of IKG. Figure 6.1 shows a representation of *Alice*'s IKG which contains the assertions described earlier as well as their attributed class.

Definition 6.2.1 (Individual's Knowledge Graph (IKG))**.** An Individual's Knowledge Graph (IKG) is the set of all assertions that share the same individual as a `subject` or `sourceIndividual`. Let i be an individual of an ontology. The IKG of i noted I in the ontology $\mathcal{O}$ is defined in Equation (6.13).

$$I = \big\{\big(i - \texttt{:predicate} - object\big) \in \mathcal{O}\big\} \tag{6.13}$$

An IKG can be represented as a star graph where each node is an individual, the center node is i and the edges represent the predicates.

In Section 6.1.1 we have seen that counterfactual explanations answer the question "What should be changed to get Q instead of P". In the context of ontologies, P and Q are classes or sets of classes of an individual e.g. an individual is of class `Denied` and wants to know the required modifications to be of class `Approved`. Resulting from this observation, Definition 6.2.2 provides the definition of a counterfactual in the context of ontologies.

Definition 6.2.2 (Counterfactual for OWL ontologies)**.** Let $\mathcal{O}$ be a consistent ontology, $i \in \mathcal{O}$ an individual and I its IKG. Given C_o, the subset of classes to modify and C_f the *foil* subset of classes (i.e. the desired subset of classes) such that $C_o \subseteq I$ and $C_f \nsubseteq I$. The IKG $\widehat{I}$ is a counterfactual of I if $C_f \subseteq \widehat{I}$ and $C_o \nsubseteq \widehat{I}$.

We have defined a counterfactual explanation for OWL ontologies. The desired properties of a counterfactual identified in Section 6.1.1 hold for any type of counterfactual. However, there is no definition of these properties adapted to OWL ontologies. Therefore, we discuss these properties and propose adequate definitions.

Validity The definition of validity for machine learning says that a counterfactual is valid if the outcome is the one expected. We add to this definition the necessity for the ontology to be consistent with the new changes. Indeed, the ontology was consistent before the changes, thus it should remain consistent with the counterfactual. Therefore, a counterfactual is valid if it respects Definition 6.2.2 and the resulting ontology is consistent.

Sparsity It is the amount of assertions modified on the original IKG to get the counterfactual.

Proximity It measures the difference between the original input and the counterfactual. Computing the closeness of two IKG is more challenging than computing the difference between two vectors. In machine learning, proximity was computed feature-wise; the equivalent for IKGs is to compute proximity assertion-wise which raises multiple issues that will be discussed later.

Plausibility Plausibility measures how realistic and achievable a counterfactual is. In machine learning, researchers were using domain knowledge to ensure that the generated counterfactuals are realistic. Ontologies are built with domain knowledge, therefore plausibility is already ensured by the consistency of an ontology and most problems faced by machine learning concerning plausibility are irrelevant. Indeed, an inconsistent counterfactual means that the counterfactual does not respect the rules given by domain experts that reflect the real world; thus rendering the counterfactual implausible.

Still, some counterfactuals could be consistent while being implausible. Some changes may not be feasible such as decreasing the age or decreasing the education level. Likewise, some assertions should not be changed as defined by actionability.

Actionability A counterfactual is actionable when it only modifies actionable assertions. An assertion is actionable if it is fair and feasible to mutate it. We propose to flag some predicates as non actionable directly in the ontology. For instance, the age should not be actionable as it is a protected feature. Therefore, the predicate `:hasAge` is marked as non actionable and any assertion containing this predicate cannot be modified.

Causality Causality checks that causal relation between assertions are respected e.g. when *Education* increases, then *Age* should also increase. Scholars used user-defined constraints to ensure these causal relations in machine learning. We argue that these causal relations should already be included in the ontology hence checking consistency also verifies causality.

Diversity Diversity is the same metric as proximity but between two counterfactuals.

These properties greatly rely on the ontology to ensure validity and plausibility. If a counterfactual is consistent with the ontology, then it is valid and most likely plausible. It is possible that a counterfactual is consistent but not plausible, which exposes an issue in the design of the ontology thus allowing the designer to debug and repair it. Sparsity is simple to compute since it corresponds to the number of modifications. Conversely, proximity and diversity are difficult to calculate. Proximity represents the difficulty of making the prescribed modifications. In machine learning, the proximity of categorical features is an open-problem since it requires domain-knowledge to accurately judge the difficulty of going from one class to another. In the context of ontologies, most assertions represent categorical features but we have domain knowledge at our disposal. Thus, a metric to calculate proximity that is specific to ontologies must be designed.

Up until now, we have simply redefined machine learning concepts to be used for ontologies, based on an example tailored for a machine learning application. However, the goals and functioning of ontologies are different from those of machine learning algorithms. The loan example

does not demonstrate all possibilities of OWL ontologies. Therefore, we propose a new example based on the Pizza ontology[2].

Example 6.2 The Pizza ontology[2] provides the definitions of multiple types of pizza and their respective toppings and bases. In this example, we study a pizza that has mozzarella, olive and chicken toppings as well as a deep pan base. This individual is called p and its corresponding IKG is P with the following assertions:

- $(p - \texttt{pizza:hasTopping} - mozzarellaTopping)$
- $(p - \texttt{pizza:hasTopping} - oliveTopping)$
- $(p - \texttt{pizza:hasTopping} - chickenTopping)$
- $(p - \texttt{pizza:hasBase} - deepPanBase)$

The class of each object corresponds to their name e.g. the individual $mozzarellaTopping$ is of class `MozzarellaTopping`, $oliveTopping$ is of class `OliveTopping` etc.

Individual p is classified as a `MeatyPizza` by the reasoner. A user may wonder what should change to classify this pizza into a `VegetarianPizza`. This calls for a counterfactual explanation. A counterfactual generated for a machine learning algorithm would modify the topping $chickenTopping$ into another type of topping that is not meat. Nevertheless, another solution made possible by ontologies is to remove this topping.

Inversely, a similar pizza without the chicken topping is classified as `VegetarianPizza` and a user wants to know the changes necessary to turn it into a `MeatyPizza`. Expected counterfactuals are to modify a topping into a meaty topping or to add a new meaty topping. Yet, with the open world assumption, nothing needs to be changed since the absence of a statement does not mean that it is false. Thus, it is possible that the presence of a meat topping is true but not known.

Example 6.2 exposes the particularities and possibilities of the generation of counterfactual for ontologies. The open world assumption enables OWL ontologies to handle missing attributes but may lead to unexpected counterfactuals e.g. no change is made to turn a vegetarian pizza into a meaty pizza. Moreover, the possibility to insert or delete assertions is desirable to generate counterfactuals as was shown in that example. This implies that a counterfactual may not have the same number of assertions as the original individual which raises new issues for the proximity and diversity metrics.

Our approach to generate counterfactuals explanations for ontologies, named the CEO method, is presented in the following section is inspired by the heuristic search approach to generate counterfactuals. First, the space of candidate counterfactuals is explored to find valid and plausible counterfactuals that have a small proximity relative to the original individual. The exploration should take into account the open world assumption and be able to insert or delete assertions. This step is computationally expensive since there may be a large amount of possible counterfactuals hence an appropriate heuristic should be used to minimize the computation cost. Then, the proximity, sparsity and diversity metrics are to be calculated to identify the best set of counterfactuals to present to the user. The proximity metric is challenging to define since the number

[2] `https://protege.stanford.edu/ontologies/pizza/pizza.owl`

of assertions between individuals will vary. Moreover, there is no pre-established method to compute the difficulty of modifying, deleting or inserting an assertion. The proposed solutions to these problems are discussed in the next section where we introduce the CEO method.

6.3 The CEO method

The CEO method takes an IKG I as input as well as the set of classes that the user wants to modify and the foil (i.e. desired) set of classes, named C_o and C_f respectively as defined in Definition 6.2.2. The output of the method is a set of valid counterfactuals chosen according to their proximity, sparsity and diversity. In Section 6.1.1, we have discussed heuristic methods to find a set of counterfactuals in the context of machine learning. We apply the same kind of procedure for the CEO method in order to find ideal candidates for counterfactuals. First, the search space for possible counterfactuals and its representation are defined. Then, a heuristic to efficiently explore this space and return valid counterfactuals is discussed. Finally, the computation of the several metrics is described and counterfactuals are chosen based on these metrics.

The search space for counterfactuals is composed of all possible individuals such that the foil set of classes is included in their IKG but not the user-defined set of classes to change. The search space does not contain counterfactuals that change non-actionable assertions. These assertions are specified directly in the ontology by making the desired properties sub-properties of `non-actionable`. Therefore, assertions where the predicate is a sub-property of `non-actionable` will be ignored when looking for counterfactuals. Let $\mathcal{O}$ be a consistent ontology, $\mathcal{C}(\mathcal{O})$ the set of IKGs resulting from all possible combinations of assertions within the ontology that are not necessarily consistent. The search space for counterfactuals Ω can be defined as follows, $\Omega = \{I \in \mathcal{C}(\mathcal{O}) | C_f \subseteq I, C_o \nsubseteq I, NA \subseteq I\}$ where NA refers to non-actionable assertions of the original individual. We have seen that IKGs are star graphs where the edges are predicates and nodes are individuals. Therefore, Ω is a set of graphs implying that a distance over graphs can be applied to calculate proximity. Specifically, we intend to use the Graph Edit Distance (GED) described in Section 6.1.3 as a measure of dissimilarity between IKGs. The search space Ω can be represented as directed graph where each node is an IKG and the arcs represent elementary operations applied to the source node to obtain the target node. Since IKGs are star graphs, only elementary operations on nodes are considered. When a node is added or removed from the IKG, the edge is subject to the same operation. We define these three elementary operations:

Assertion modification Modifies the object of an assertion without changing the predicate.

Assertion deletion Deletes an assertion from the IKG i.e. removes both the corresponding node and arc from the IKG.

Assertion insertion Inserts an assertion to the IKG i.e. adds a node containing the object of the new assertion and connects it to the center individual with an arc representing the predicate of this assertion.

Figure 6.2 represents an example of the search space, with four different IKGs and the arcs that connect them. We note that each operation has an inverse operation. Insertion and deletion operations on the same assertion are inverse of one another. Likewise, the inverse of a modification is also a modification with the change of object inverted. Therefore, Ω is a symmetric directed graph [222] i.e. for every arc (i_1, i_2) there is also an arc (i_2, i_1). With this representation, sparsity can be computed as the length of the shortest path between the original IKG and any counterfactual. Similarly, proximity is easily calculated with the GED using this representation. Nevertheless, the cost functions for each elementary operation and a heuristic to efficiently explore the search space remain to be defined.

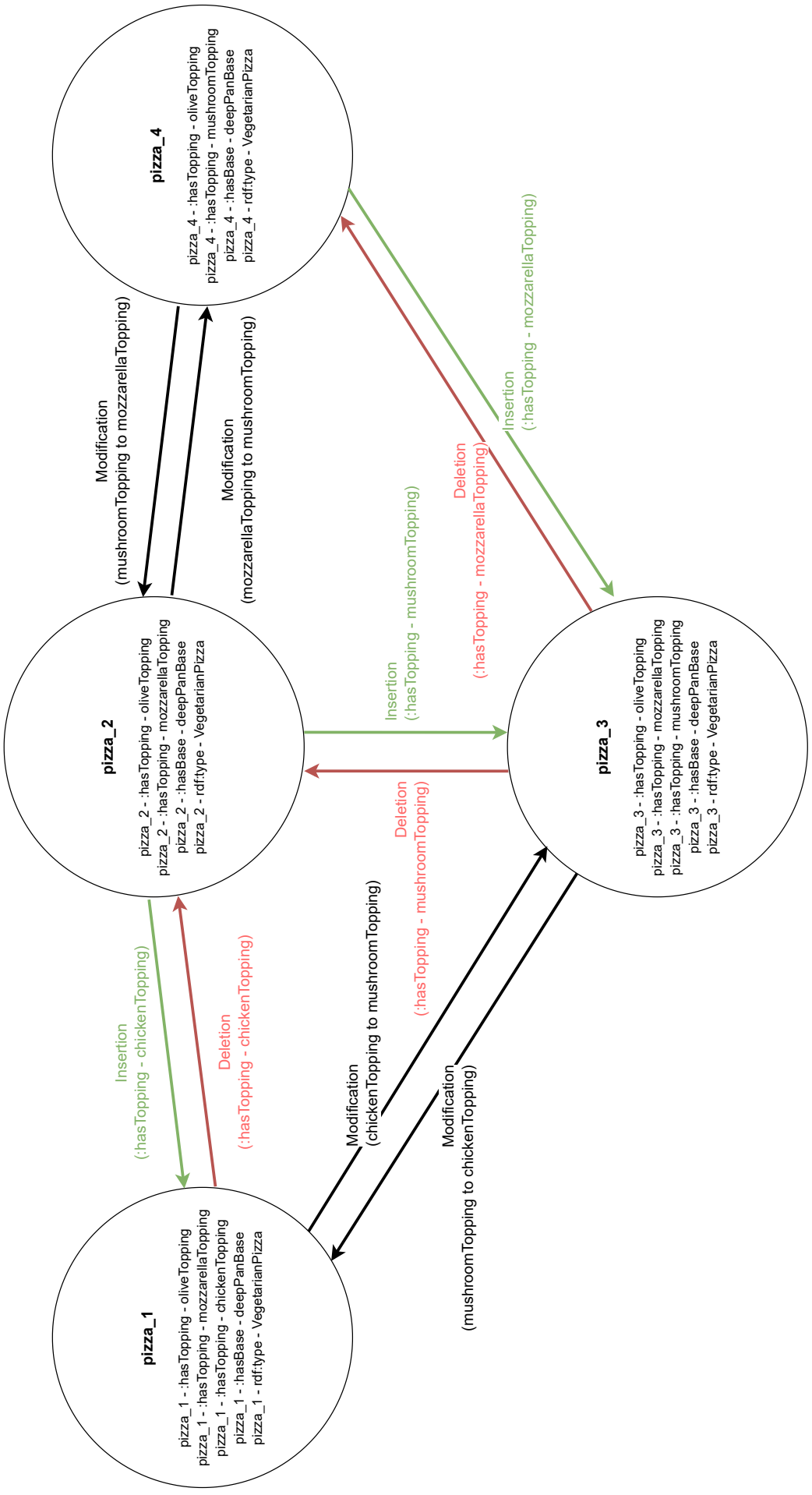


Figure 6.2: Graph representation of the search space Ω.

6.3.1 Exploring the counterfactuals space

The search space Ω contains all possible counterfactual IKGs, making it expensive to compute. Moreover, only valid IKGs are of interest and they represent a fraction of this space. We propose to explore the graph by starting from the counterfactual that is closest to the original IKG and then search for valid counterfactuals in its neighborhood. We argue that the closest counterfactual from the original IKG is the one where only its class assertions are modified to satisfy Definition 6.2.2. Let I be the original IKG and $\widehat{I}_0$ the closest counterfactual i.e. the counterfactual with a minimal amount of modifications. This counterfactual is defined as $\widehat{I}_0 = (I \setminus C_o) \cup C_f$, the set of classes to modify C_o is removed and replaced by the foil set C_f. It is unlikely that $\widehat{I}_0$ is a valid counterfactual, therefore the search space is explored by applying elementary operations to $\widehat{I}_0$.

The ontology is considered consistent with the original IKG I. This individual is replaced by the counterfactual $\widehat{I}_0$ and the consistency of the ontology is tested. Hence, any inconsistency detected by the logical reasoner comes solely from this counterfactual. We assume that the foil set of classes C_f does not create any inconsistency on its own i.e. C_f does not contain contradictory classes such as $\{MeatyPizza, VegetarianPizza\}$. Thus, the inconsistency comes from at least one `ObjectPropertyAssertion` or `DataPropertyAssertion` in the IKG. The objective of the following heuristic is to identify the faulty assertions and either delete or modify them. We define a *faulty* assertion as *an assertion that provokes an inconsistency in an ontology*. For instance, the assertion $(veggiePizza - \texttt{:hasTopping} - chickenTopping)$ is a faulty assertion when $veggiePizza$ is a `VegetarianPizza`.

The heuristic is composed of three steps: detection of faulty assertions, generation of ancestors and generation of descendants. The detection of faulty assertions explores the graph by deleting some assertions of $\widehat{I}_0$ until valid IKGs are found. Once these faulty assertions have been identified, modifications on the objects of these assertions are attempted in order to find consistent IKGs that retain these assertions. Ancestor classes of the object of each assertion are traversed until one ancestor class produces a consistent counterfactual; this is the ancestor generation phase. Then a similar phase seeks for descendant classes of the consistent ancestor that also produces consistent counterfactuals; this is the descendant generation phase. At each phase of this heuristic, the explored IKGs are added to a graph Ω' that is a reconstruction of the search space Ω.

Faulty assertions detection

The first step consists in exploring the graph by deleting the assertions from $\widehat{I}_0$ one by one, in any order, until consistent IKGs are found. The deletion of a faulty assertion returns a consistent IKG thanks to the open world assumption. Indeed, with this assumption, deleting an assertion does not imply that an assertion with the same predicate does not exist. Instead, the reasoner assumes that the necessary assertions exist but are not made explicit. For instance, a pizza classified as `MeatyPizza` but with no explicit topping in its IKG is consistent because the reasoner assumes that a meaty topping may be present in the individual. The resulting valid counterfactuals can then be expanded by inserting back the faulty assertions with modifications on the objects to remove the inconsistency.

This step is described in Algorithm 6.1. The function takes an IKG as input, creates a new graph only containing the input and checks its consistency. If it is not consistent, new IKGs are generated resulting from the deletion of assertions of the input. The resulting IKGs are added to the graph and the function is recursively called on each of these new IKGs until consistent IKGs are found. Once the function ends, it returns a subgraph of Ω and all the produced subgraphs are connected together through a graph composition operation.

Figure 6.3 illustrates this process, starting with the IKG $\widehat{I}_0$ composed of three assertions that are not `ClassAssertions`. In this example, the assertion $(\widehat{I}_0 - p_3 - o_3)$ is responsible for the inconsistency. Each assertion is deleted resulting in the creation of three new IKGs $\widehat{I}_1^1$, $\widehat{I}_1^2$ and $\widehat{I}_1^3$. The function is called on these new IKGs, but stops for $\widehat{I}_1^3$ since it is consistent. For the two other IKGs, the exploration continues. We observe that this method guarantees that at least one

Algorithm 6.1 Initial exploration algorithm to identify faulty assertions.

function EXPLOREGRAPH(I)
  $\Omega \leftarrow (\{I\}, \{\})$ ▷ Create a new graph where the only node is the input.
  if $isConsistent(I)$ **then**
    return Ω
  end if
  for $A_i \in I$ **do** ▷ For all assertions $A_i \in I$, except class assertions.
    $I' \leftarrow I \setminus A_i$
    $\Omega \leftarrow compose(\Omega, exploreGraph(I'))$ ▷ Composes the resulting graphs and adds the edges between the generated IKGs.
  end for
  return Ω
end function

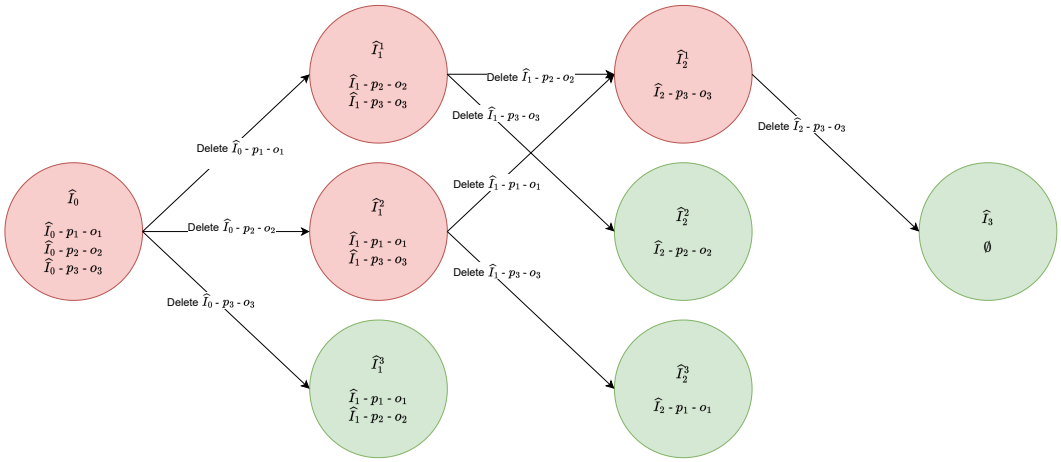


Figure 6.3: Process of exploring the search space by deleting faulty assertions. Nodes in red are inconsistent, nodes in green are consistent.

consistent IKG is found. In the worst case, an IKG with no assertion is explored and is necessarily consistent as is the case with the IKG $\widehat{I}_3$. We also note that the assertion that was deleted prior to finding a consistent IKG is a faulty assertion.

An alternative heuristic exploits the output of the reasoner to detect faulty assertions. Therefore this exploration step is less costly since faulty assertions are directly targeted and deleted. As a result, less IKGs are explored in the graph which leads to a decrease in computation time but also a decrease in the diversity of the counterfactuals. In practice, this heuristic is dependent on the reasoner's output. We have observed that in many cases, the reasoner does not provide a reason for an inconsistency which prevents the application of this alternative heuristic.

Generation of ancestors

Once the graph is explored and faulty assertions are detected, the CEO method adds back the faulty assertions. The objects of these assertions are modified by exploring their ancestor (or parent) classes until one ancestor class leads to a consistent individual. Algorithm 6.2 describes the functioning of this step. Given an IKG and a known faulty assertion, this algorithm attempts to insert this assertion back, retaining the original predicate but modifying the object. The original object is modified into its ancestors i.e. parent classes of this object are explored until either a consistent parent is found or the root parent is reached. If the root parent is reached and the IKG is not consistent, it means that the predicate of the faulty assertion is the problem and it cannot be inserted back. Usually, the root parent corresponds to either `owl.Thing` or the declared `range` of the predicate.

Algorithm 6.2 Ancestors generation algorithm

function GENERATEANCESTORS(I, A_f) ▷ A_f is the faulty assertion to insert back.
$\Omega \leftarrow (\{I\}, \{\})$
$pred \leftarrow predicate(A_f)$
$o \leftarrow object(A_f)$
$p \leftarrow Parents(o)$ ▷ $Parents(o)$ returns the set of instance of every class that is a direct parent of individual o. We note $p = \{p_0, p_1, \ldots, p_n\}$ where p_k is an instance of the k-th parent class of o.
$consistent \leftarrow False$
while not $consistent$ and $p \neq \emptyset$ **do**
for $k \leftarrow 0$ to $|p|$ **do** ▷ $|p|$ is the size of p.
$I' \leftarrow I$
$I' \leftarrow I' \cup (i' - pred - p_k)$ ▷ i' is the center individual of the IKG I'.
$\Omega \leftarrow addNode(I', \Omega)$ ▷ Add I' and corresponding edges to Ω.
if $isConsistent(I')$ **then**
$consistent \leftarrow True$ ▷ When at least one individual in the parents is consistent, the while loop stops.
end if
end for
$p \leftarrow \bigcup_{i=0}^{|p|} Parents(p_i)$
end while
return Ω
end function

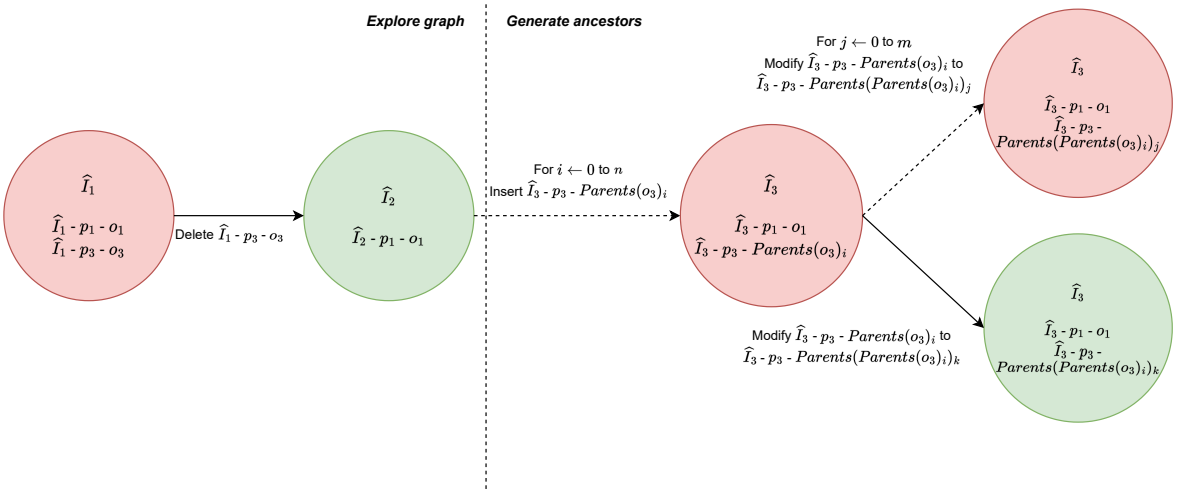


Figure 6.4: Process of generating ancestors for one individual knowing the faulty assertion. Nodes in red are inconsistent, nodes in green are consistent, dotted lines represent edges between one node to a list of other nodes. The faulty assertion is identified thanks to the $exploreGraph$ function.

Figure 6.4 depicts the functioning of this algorithm based on the output of the $exploreGraph$ function described before. The $exploreGraph$ found a consistent IKG by deleting the assertion with the predicate p_3 and the object o_3. Therefore this assertion is faulty and $generateAncestors$ inserts it back and modifies the object into one of its parents class. The function $Parents$ returns the set of instance of every class that is a direct parent of the individual given in input. The output of this function is a list of individual such that the class of each individual in the list is a parent class of the input. For example, let $vegetarianPizza$ be an instance of the class `VegetarianPizza`. The function $Parents(vegetarianPizza)$ returns a set of the instances of ev-

ery direct parent class of `VegetarianPizza` which is only the class `Pizza` in this case. Thus $Parents(vegetarianPizza) = \{pizza\}$ where $pizza$ is an individual of class `Pizza`. For each individual returned by $Parents$, a new IKG is created and its consistency is tested. In Figure 6.4, the direct parents of o_3 were not consistent, so the second order ancestors (parents of parents) are explored and one consistent IKG is found so the function stops.

The algorithm $generateAncestors$ is run for every consistent IKG present in the graph after the initial exploration phase described before. The faulty assertion is found by looking for adjacent inconsistent nodes connected to the IKG of interest with a deletion operation. The assertion deleted in this operation is used as the faulty assertion. In the case where an object has multiple parent classes, each parent class is explored separately. After generating these ancestors, the same process is applied to find consistent descendants.

Generation of descendants

The last step is to generate descendants of all consistent IKGs in the graph. Algorithm 6.3 shows that the functioning of this process is similar to the $generateAncestors$ algorithm. The subclasses or descendants of an individual o are obtained with the function D. This function returns a list of direct descendants of o noted d_i in the algorithm. IKGs are generated by replacing the object of the target assertion by d_i for every descendant. Like $generateAncestors$, the consistency of these IKGs is tested and $generateDescendants$ is called again on the resulting consistent IKGs.

Algorithm 6.3 Descendants generation algorithm

function GENERATEDESCENDANTS(I, A) ▷ A is an assertion of I, I is consistent.
  $\Omega \leftarrow (\{I\}, \{\})$
  $pred \leftarrow predicate(A)$
  $o \leftarrow object(A)$
  $c \leftarrow Children(o)$ ▷ Like $Parents$, $Children$ returns the set of instance of every class that is a direct child of individual o. We note $c = \{c_0, \ldots, c_n\}$ where c_k is an instance of the $k-th$ child class of o.
  for $k \leftarrow 0$ to $|c|$ **do**
    $I' \leftarrow I \setminus A$
    $A' \leftarrow (i' - pred - c_i)$ ▷ i' is the center individual of the IKG I'.
    $I' \leftarrow I' \cup A'$
    $consistent \leftarrow isConsistent(I')$
    $\Omega \leftarrow addNode(I', \Omega)$
    if $consistent$ **then**
      $\Omega \leftarrow compose(\Omega, generateDescendants(I', A'))$
    end if
  end for
  return Ω
end function

Figure 6.5 illustrates the $generateDescendants$ function, following the example given in Figure 6.4. In the previous step, an ancestor of o_3 was found to be consistent that we note o. The generation of descendants seeks to find descendants of this object that are consistent to get as close as possible to the original IKG $\widehat{I}_0$. The $Children$ function works the same as $Parents$ and returns the set of individual that are instances of all children classes of the input. In the example, the classes of individual o have n descendants, the i-th descendant is represented as c_i. All the descendants are inconsistent except for the k-th descendant. The $generateDescendants$ function is then called on this k-th descendant to further explore the graph with consistent IKGs. This process is repeated until new there are no more descendants or no descendant IKG is consistent.

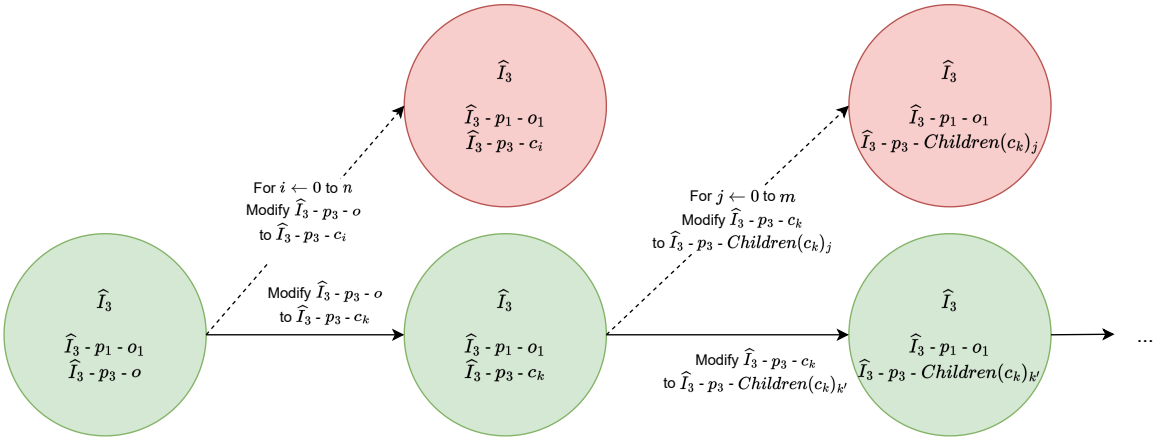


Figure 6.5: Process of generating descendants for one individual. Nodes in red are inconsistent, nodes in green are consistent, dotted lines represent edges between one node to a list of other nodes.

Algorithm 6.3 is applied to every consistent individual of the graph. The goal is to find valid counterfactuals that repair faulty assertions with objects that are similar to the original objects i.e. replace the faulty object with an object that has the same level of abstraction. For example, to replace $chickenTopping$, another specific type of topping may be expected such as $artichoke\ Topping$ or $fourCheesesTopping$. Restricting the search to ancestors may lead to counterfactuals limited to abstract classes e.g $VegetableTopping$ or $PizzaTopping$. This step of the heuristic allows the identification of a large diversity of counterfactuals.

Counterfactual generation heuristic

Algorithm 6.4 summarizes the heuristic used to generate counterfactuals. First, the closest counterfactual from the original input $\widehat{I}_0$ is built and is the starting node of the graph Ω. The rest of the algorithm is executed if $\widehat{I}_0$ is not consistent, otherwise it is returned as the only counterfactual since it is considered the closest to the original input. The second step is to explore the graph by deleting assertions from $\widehat{I}_0$ until consistent IKGs are found, with the $exploreGraph$ function. At this point, every consistent IKG in the graph is the result of at least one deletion operation. A faulty assertion is the last assertion deleted to obtain a consistent IKG. A new set of IKGs are generated with $generateAncestors$ by inserting and fixing these faulty assertions. Finally, descendants of every consistent IKG in the graph are explored with $generateDescendants$ to increase diversity and prevent counterfactuals that are too abstract as discussed earlier.

This heuristic guarantees that at least one counterfactual is found as well as encourages the generation of many diverse counterfactuals. It does not use the information provided by the ontology or the reasoner to guide the search of counterfactuals. This is useful for debugging the ontology since it is not biased by the ontology's definitions and may generate unexpected counterfactuals that should not be possible. In return, this blind exploration scales exponentially with the number of assertions in the original IKG. Moreover, it is not capable of inserting new assertions that were not present in the initial individual. A way to solve this last issue is to explicitly declare that a deleted assertion cannot be present in the IKG. Hence, the reasoner will not infer that the assertion is implicitly present because of the open world assumption. The insertion of a new assertion may therefore be required to get a consistent counterfactual. Yet, this solution calls for even more exploration by inserting new assertions, adding more complexity to the current heuristic. In addition, the current heuristic relies on the fact that an individual with no assertion is inherently consistent. The proposed solution would break this assumption and break the heuristic. Finally, this heuristic does not handle `DataPropertyAssertions` because the possible modifications are infinite compared to `ObjectPropertyAssertions` where the modifications are limited to the classes of the ontology.

Algorithm 6.4 Heuristic to generate counterfactuals for ontologies

function GENERATECOUNTERFACTUALS(I, C_o, C_f)
  $\widehat{I}_0 \leftarrow (I \setminus C_o) \cup C_f$
  $\Omega \leftarrow \left(\{\widehat{I}_0\}, \{\}\right)$
  $\Omega \leftarrow compose\left(\Omega, exploreGraph\left(\widehat{I}_0\right)\right)$
  for $\widehat{I}_i^c$ in Ω **do** ▷ $\widehat{I}_i^c$ is the i-th consistent IKG in Ω
    $A_f \leftarrow getFaultyAssertion\left(\widehat{I}_i^c, \Omega\right)$
    $\Omega_i \leftarrow generateAncestors\left(\widehat{I}_i^c, A_f\right)$
  end for
  $\Omega \leftarrow compose(\Omega, \Omega_1, \ldots, \Omega_n)$
  for $\widehat{I}_i^c$ in Ω **do**
    for A_i in $\widehat{I}_i^c$ **do**
      $\Omega_i \leftarrow generateDescendants\left(\widehat{I}_i^c, A_i\right)$
    end for
  end for
  $\Omega \leftarrow compose(\Omega, \Omega_1, \ldots, \Omega_n)$
  return Ω
end function

6.3.2 Computing metrics

The CEO method represents the space of possible counterfactuals as a symmetric directed graph [222] where nodes are IKGs. The edges represent three elementary operations on an IKG's assertions: modification, deletion and insertion. This structure enables the use of a graph edit distance to compute the dissimilarity between two IKGs and therefore to compute proximity and diversity. Graph edit distances require the definition of cost functions that represent the cost of applying each elementary operation.

Concerning deletion and insertion operations, we have not found methods that could compute the cost of such operations. Moreover, counterfactuals for machine learning do not handle these possibilities. We argue that deleting an assertion removes any information about the changes to do. With the open world assumption, deleting an assertion is equivalent to either modifying the object into an unknown class or completely removing the assertion. As a result, there is no more information on the modifications that need to be made to attain the counterfactual, leading to changes that are incomprehensible to the user. Similarly, inserting an assertion requires the user to achieve something that was absent from their starting point. The reasons for the absence of a certain assertion are unknown and inserting this assertion may also lead to incomprehensible or unfeasible results. Moreover, inserting an assertion adds uncertain information which is arguably worse than removing information. Modification operations also add uncertainty but in a less impactful way than insertion since modification assures that the predicate was already present in the original IKG. From this discussion, we propose that for a given assertion, the cost of insertion should be greater than the cost of deletion and similarly, the cost deletion should be greater than the cost of modification. Equation (6.14) is the proposed cost function, where sim is a similarity measure for two assertions and $\mathcal{A}$ is the set of all possible assertions. This formulation reduces the problem to defining a dissimilarity measure between two assertions.

$$c(e) = \begin{cases} sim(a_1, a_2) & \text{if } e \text{ is a class modification operation from assertions } a_1 \text{ to } a_2 \\ \max\limits_{a \in \mathcal{A}} sim(a_1, a) + 1 & \text{if } e \text{ is a deletion operation where } a_1 \text{ is the studied assertion.} \\ \max\limits_{a \in \mathcal{A}} sim(a_1, a) + 2 & \text{if } e \text{ is an insertion operation where } a_1 \text{ is the studied assertion.} \end{cases} \tag{6.14}$$

Computing the similarity between two assertions is a problem that is well documented in the literature as discussed in Section 6.1.3. Fernández-Chamizo et al. [216] proposed a similarity measure between two individuals. This similarity measure contains a similarity function between

assertions of the two individuals. They stated that only the assertions with the same predicate can be compared. Indeed, the predicate of an assertion represents the nature of the relation e.g. `rdf:type` or `:hasTopping`. It is clear that assertions of different nature should not be compared. Fernández-Chamizo et al. compared relations by comparing the sets of individuals that are object of the predicate. This differs from our goal that is to compare assertions, thus we do not have sets of individuals. Nevertheless, we simplify the problem of comparing assertions to the problem of comparing the objects of these assertions.

Most solutions to compare two individuals compare the concepts of which these individuals are instances. Several similarity measures between concepts or classes have been proposed. However, we discussed that these measures apply recursive calls that can infinitely loop if the concept definitions are cyclical. The goal of the CEO method is to be compatible with any ontology, thus preventing the use of these measures. Similarly, we cannot assume that a set of unbiased individuals of the ontology is accessible, also preventing semantic similarity measures. These constraints motivated the choice of the edge-counting similarity measure [219]. The `SubClassOf` relations are used to generate a hierarchy tree where classes of the ontology are the nodes. The root of this hierarchy is the class `owl.Thing`. Thus, the similarity between two classes is the length of the shortest path from one class to the other in this tree.

We are now able to compute the proximity, diversity and sparsity for any counterfactual in the graph. Equation (6.15) shows the formula to compute proximity, where $\mathcal{P}\left(\widehat{I}_0, \widehat{I}\right)$ is the set of paths i.e. the set of lists of elementary operations to go from $\widehat{I}_0$ to $\widehat{I}$; c is the cost function defined in Equation (6.14) with the edge-counting method and w is a constant that represents the similarity between the original individual I and its closest counterfactual $\widehat{I}_0$. For simplicity, we set $w = 0$ since it does not affect the results.

$$proximity\left(I, \widehat{I}\right) = w + \min_{(e_1,\ldots,e_n)\in\mathcal{P}\left(\widehat{I}_0,\widehat{I}\right)} \sum_{i=1}^{n} c(e_i) \tag{6.15}$$

Sparsity is computed as the length of the shortest path between $\widehat{I}_0$ and another counterfactual. Finally, diversity between two counterfactuals is computed in the same way as proximity with small differences such as the absence of the constant w, as shown in Equation (6.16).

$$diversity(I_1, I_2) = \min_{(e_1,\ldots,e_n)\in\mathcal{P}(I_1,I_2)} \sum_{i=1}^{n} c(e_i) \tag{6.16}$$

The main advantage of the graph edit distance to compute proximity is that any cost function can be applied without modifying the method. Our definition of cost functions are made to be easily applicable to any ontology but may not be precise enough. There is no ideal way to compute this similarity and the choice should be specific to each ontology and task. For instance, ontologies that dispose of a set of instances should rather use semantic metrics instead of syntactic ones. These metrics are used to select and rank the counterfactuals based on their proximity, sparsity and diversity. The selected counterfactuals are presented to the user from top to bottom ranking (lowest to highest proximity and sparsity). In the next section, we apply the CEO method to Example 6.2 in order to validate the approach and identify its shortcomings. A preliminary user-study to further evaluate this method is proposed in Chapter 7.

6.4 Validation

The validation of the CEO method evaluates the quality of the counterfactuals and the execution time of the algorithm. In this section, we focus on the execution time and whether the method returns the expected results. The quality of the generated counterfactuals will be studied and discussed in Chapter 7 with a user-study. The use-case is based on Example 6.2 which uses the Pizza ontology[3].

[3] `https://protege.stanford.edu/ontologies/pizza/pizza.owl`

6.4.1 Test cases and results

Let $C_o = \{MeatyPizza\}$, $C_f = \{VegetarianPizza\}$, A_f a set of faulty assertions that will produce inconsistencies with C_f and A_c a set of correct assertions that will not lead to inconsistencies. The CEO method is applied to generate a set of valid counterfactuals, with $I = C_o \cup A_f \cup A_c$ as the input IKG, C_o as the set of classes to change and C_f as the foil set of classes. The goal is to understand how the CEO method behaves with a different number of faulty and correct assertions. Specifically, we monitor the running time of each step of the heuristic in relation to the number of candidate counterfactuals explored. Moreover, we declare some counterfactuals that we expect e.g. change meat topping to vegetable topping. The rank of these expected counterfactuals is monitored to verify that they are generated by the method and are among the counterfactuals with the lowest proximity.

Case 1: One faulty assertion

The first case is an individual that has only one assertion that is faulty.

$$A_c = \emptyset$$
$$A_f = \{(i - \texttt{:hasTopping} - chickenTopping)\}$$

We expect the counterfactuals to modify $chickenTopping$ into $vegetableTopping$ or $cheese\ Topping$. The CEO method generated 41 valid counterfactuals out of 52 explored candidates. The proximity of the expected results was tied with 5 other counterfactuals and was the second smallest proximity. The counterfactual with the smallest proximity modifies $chickenTopping$ to $pizzaTopping$ which could represent any topping. The second smallest proximity corresponds to a modification of $chickenTopping$ to direct subclasses of `PizzaTopping` e.g. `CheeseTopping`, `FruitTopping` or `VegetableTopping`. The counterfactual with the highest proximity is the only one that deletes the assertion.

Case 2: One faulty and one correct assertions

This second example adds a correct assertion to the previous case.

$$A_c = \{(i - \texttt{:hasTopping} - mozzarellaTopping)\}$$
$$A_f = \{(i - \texttt{:hasTopping} - chickenTopping)\}$$

We expect the counterfactuals to leave the assertion in A_c untouched and to modify $chicken\ Topping$ into $vegetableTopping$ or $cheeseTopping$. The CEO method generated 348 valid counterfactuals out of 911 explored candidates. We note that the amount of explored and valid counterfactuals drastically increased compared to the previous case. The ranking is similar to the last case for the first 41 counterfactuals, meaning that the expected counterfactuals ended up at the same rank with the same proximity. The CEO method tested almost every combination of pairs of toppings resulting in this increase in the number of counterfactuals generated.

Case 3: Two faulty assertions

In order to assess the impact of faulty assertions, we propose an example that also has two assertions similar to the previous one, but both assertions are now faulty. Therefore, we set A_c and A_f as follows.

$$A_c = \emptyset$$
$$A_f = \{(i - \texttt{:hasTopping} - chickenTopping), (i - \texttt{:hasTopping} - hamTopping)\}$$

We expect the counterfactuals to change both assertions into any combination of $vegetable\ Topping$ and $cheeseTopping$. The CEO method generated 317 valid counterfactuals out of 1160

explored candidates. Compared to the previous case, more counterfactuals were explored but less were valid. It can be explained by the increased difficulty to get the individual consistent since both assertions are faulty. The number of explored counterfactuals did not change significantly because the assertions are of the same nature resulting in the same combinations to explore. The first 40 counterfactuals in the ranking have modified one topping into the abstract class $pizzaTopping$ while exploring every valid topping on the other assertion. Thus, our expected counterfactuals are not in these 40 counterfactuals. The rest of the assertions are similar to the last case, with every valid combination of two toppings.

Case 4: One faulty topping, one faulty base

We propose an additional test case that differs from the other in the nature of the assertions. The aim is to test the influence of the predicate on the number of counterfactuals explored.

$$C_f = \{VegetarianPizza, RealItalianPizza\}$$
$$A_c = \emptyset$$
$$A_f = \{(i - \texttt{:hasBase} - deepPanBase), (i - \texttt{:hasTopping} - chickenTopping)\}$$

The `RealItalianPizza` class imposes that the base is of class `ThinAndCrispyBase`. The number of counterfactuals explored is proportional to the number of classes in the range of a predicate. The predicate *:hasBase* has only three classes in its range. Therefore, a decrease in the number of explored counterfactuals should be observed. The expected counterfactual is the modification of $deepPanBase$ to $thinAndCrispyBase$ and the modification of $chickenTopping$ to $vegetableTopping$.

The CEO method generated 123 valid counterfactuals out of 208 nodes explored which validates our expectations. The 10 best ranked counterfactuals modify the topping with different classes while always changing the base from $deepPanBase$ to the abstract $pizzaBase$, except for the tenth counterfactuals which changes the base to the expected $thinAndCrispyBase$. Then, every possible combination of base and topping is generated. Abstract classes (e.g. `PizzaTopping` or `PizzaBase`) are attributed a lower proximity which favors them in the ranking. The expected counterfactual is found by the method but ranked at the 42nd position.

Finally, we note that running these examples allowed us to identify design issues in the Pizza ontology[4]. These issues are intentional as the initial goal of the Pizza ontology is to act as a tutorial that highlights typical design errors that can be made when building an ontology. Nevertheless, the CEO method showed its ability to detect such issues.

6.4.2 Analysis

The experiments showed that the expected counterfactuals are always generated but not always well ranked. The computation of proximity clearly favors these abstract classes as they systematically have the lowest proximity and thus highest rankings. This is due to the computation of proximity which uses the edge-counting dissimilarity measure. This dissimilarity penalizes classes that have the same level of abstraction. For instance, the dissimilarity between the classes `ParmezanTopping` and `MozzarellaTopping` is 2. Yet, they are are both direct subclasses of `CheeseTopping` which should make them highly similar. The edge-counting dissimilarity favors parent classes, the dissimilarity between `CheeseTopping` and any direct subclass is always 1. As a result, abstract classes lead to low proximity.

Consequently, the top counterfactual is always too abstract to the point that it does not provide any relevant information. In the studied cases, the top counterfactual always modified the faulty assertions to $pizzaTopping$, meaning that the meat should be replaced by anything. Users that are not experts in ontologies or that do not know the functioning of the CEO method may be confused by this counterfactual. Giving the information that meaty toppings are problematic and

[4] `https://protege.stanford.edu/ontologies/pizza/pizza.owl`

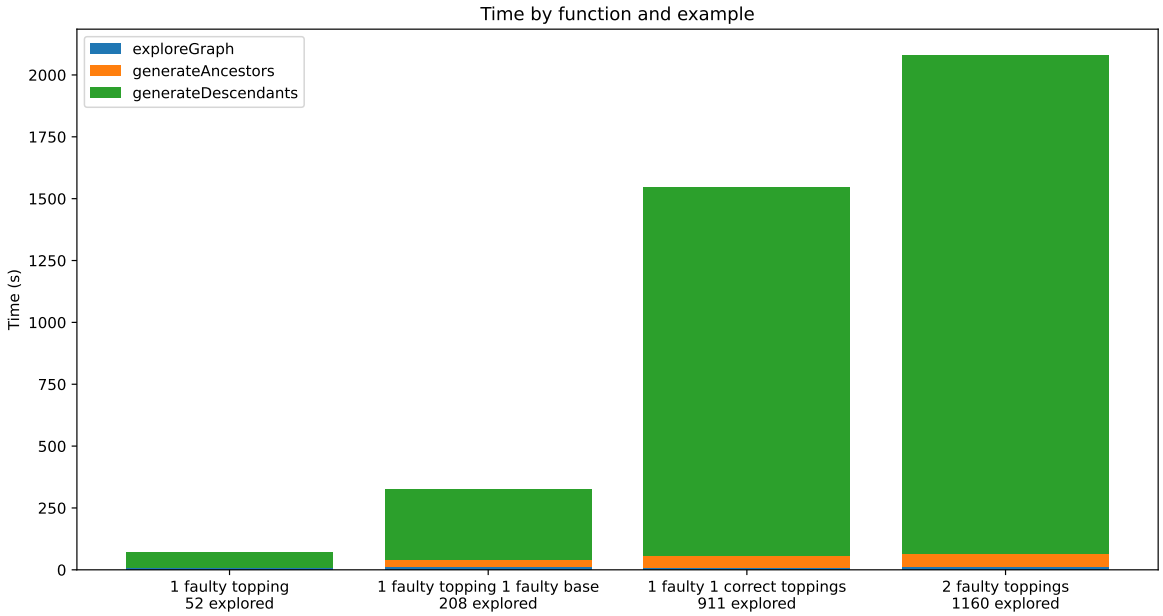


Figure 6.6: Bar plot of the detailed execution time for each example.

should be replaced with any topping that isn't meaty may be more valuable to the user. This observation echoes with the discussion that concludes Section 6.3.1, modifications or deletions that led to inconsistencies should be taken into account in the exploration and presented to the user.

The execution time observed in these test cases is not satisfactory. Figure 6.6 clearly demonstrates that the last step of the CEO method represents approximately 90% of the total execution time. The execution time is tightly linked to the number of explored nodes. The heuristic seeks every consistent combination of assertion modifications. The number of possible modifications for one assertion is dictated by the number of defined classes within the `range` of the assertion's predicate. In Appendix A, we pose the formula to calculate the size of the search space based on the number of classes within the `range` of each assertion's predicate. Cases 3 and 4 illustrate the consequence of the size of the search space on the number of explored nodes The `range` of the predicate *:hasBase* contains 3 classes, while the `range` of *:hasTopping* has 52 classes. The size of the search space for case 3 is 2809 while for case 4 it has a size of 212. The difference in search space explains the difference of scale in the execution time. More than 90% of this time is spent by the logical reasoner[5] which is called for every new counterfactual.

Overall, the CEO method is validated in the sense that it generates valid counterfactuals including the expected ones. Nevertheless, it faces the same problem as heuristic-based machine learning counterfactual methods i.e. a high execution time due to large search spaces. In its current state, the CEO method does not take diversity into account when exploring the search space. Maximizing diversity might be a way to decrease the number of explored counterfactuals without hindering the quality of the proposed counterfactuals. Regarding the execution time, further investigations to reduce it should be conducted. A possible direction is to explore ways to reduce the number of calls to the logical reasoner and optimize the execution time of the logical reasoner. The heuristic may also be tuned to limit the number of counterfactuals explored. For instance, the user may choose to ignore certain classes to decrease the size of the search space e.g. ignore leaf classes such as `ArtichokeTopping` or `CaperTopping` to focus on their parent class `VegetableTopping`. We have already implemented options to avoid exploring the entire search space to decrease the execution time at the expense of exhaustiveness.

[5]We used the Pellet reasoner for the experiments.

6.5 Conclusion

We have proposed the CEO method, an explainability method dedicated to ontologies that generates counterfactual explanations. Its design is inspired from XAI techniques that generate counterfactuals for machine learning. The inputs of the method are the foil set of classes and an IKG i.e. the set of assertions that share the same individual as the subject. A counterfactual represents a set of modifications on the input IKG to change its original set of classes to the foil set of classes in a way that is consistent with the ontology. First, a graph of candidate counterfactuals is explored following a specific heuristic. Then, these counterfactuals are filtered to keep only the valid and feasible ones. Afterwards, proximity and sparsity metrics are computed to identify the best counterfactuals i.e. the ones that minimize proximity and sparsity. Each step is independent from one another. This makes the CEO method highly modular and compatible with most ontologies. Choices for each step must be made based on the application. For instance, a tradeoff must be made between execution time and the number of generated counterfactuals regarding the first step. Likewise, the proximity metric requires an adapted similarity metric to adequately identify the best counterfactuals. This modularity is both an advantage and a drawback. Indeed, the CEO method can be tailored to each user which is encouraged to improve explainability. However, it requires to make informed choices for each step which makes it complex to implement.

We have tested this method on the Pizza ontology[6] to verify that the method behaves as expected and produces valid counterfactuals. The CEO method generated good counterfactuals but the experiment revealed that its algorithmic complexity is unsatisfactory and the method is not scalable. The current heuristic favors a deep exploration of the search space rather than minimizing the number of explored counterfactuals. We note that the execution time does not scale linearly with the size of the search space. In addition, the heuristic is not yet capable of handling the insertion operation and certain assertion types (e.g. `DataPropertyAssertion` or `NegativeObjectPropertyAssertion`). This limits the compatibility of CEO with certain ontologies and hinders its ability to identify some counterfactuals. Finally, we observed that the proximity metric favors the most abstract modifications. We stated that explanations that are too abstract may not be valuable to the user.

The CEO method requires some further work to address the aforementioned issues. The algorithmic complexity can be improved in two ways, either by optimizing the current algorithm or by modifying the heuristic search. In any case, the heuristic search needs to be reworked to be able to handle all types of assertions as well as insertion operations. Different heuristics may be developed with different properties e.g. favor execution time over diversity or favor insertion operations. We hypothesized that the proximity metric is not ideal for most users. The literature review highlighted that such intuition requires to be verified with an actual user study. Therefore, we propose a user study in Chapter 7 to evaluate the quality and relevance of the explanations generated by CEO and also test the hypothesis about the proximity. We further discuss possible improvements and further work on CEO at the end of the user study.

[6] `https://protege.stanford.edu/ontologies/pizza/pizza.owl`

Chapter 7

Experiments

Contents

7.1 Evaluation task . . . 127
7.1.1 The musical instruments ontology . . . 128
7.1.2 The musical instruments dataset . . . 129
7.2 Experiments on OBIC . . . 130
7.2.1 Methodology . . . 130
7.2.2 OBIC implementation . . . 131
7.2.3 Results . . . 132
7.2.4 Analysis . . . 135
7.3 Experiments on the counterfactual explanations . . . 138
7.3.1 Methodology . . . 139
7.3.2 Results . . . 140
7.3.3 Analysis . . . 142
7.4 Conclusion . . . 144

In this thesis, we have proposed a design for a complete XIS using the OBIC framework as an explainable system capable of detecting errors in the predictions using an ontology. Then, as a follow-up to this contribution, we introduced the CEO method that generates counterfactual explanations for ontologies. The CEO method was designed to provide further explanations concerning the error detection system from OBIC. In this chapter, we evaluate these contributions on the same task, that is presented in Section 7.1. Then, the performance of the error detection of OBIC is evaluated in Section 7.2. Afterwards, the CEO method is evaluated with a small scale user-study in Section 7.3. Finally, the results of these evaluations are discussed in Section 7.4.

7.1 Evaluation task

We evaluate our contributions on a musical instrument classification task. Images of musical instruments are given as input of our XIS and the goal is to determine which instrument is present. This particular task was chosen because musical instruments have particular visible characteristics that make it possible to distinguish one instrument from another. We created a novel ontology of musical instruments that leverages these observable characteristics to define the instruments. This task is well suited to evaluate the OBIC framework as it requires the use of observable properties in the class definitions. We further discuss the design of this ontology in Section 7.1.1. There was no dataset that matched with the classes defined in the ontology. Therefore, a dataset of images of musical instruments was automatically built by parsing Google Images for pictures of musical instruments based on the classes of the ontology. We manually verified the quality and correct annotations for the images. Details about the dataset are provided in Section 7.1.2.

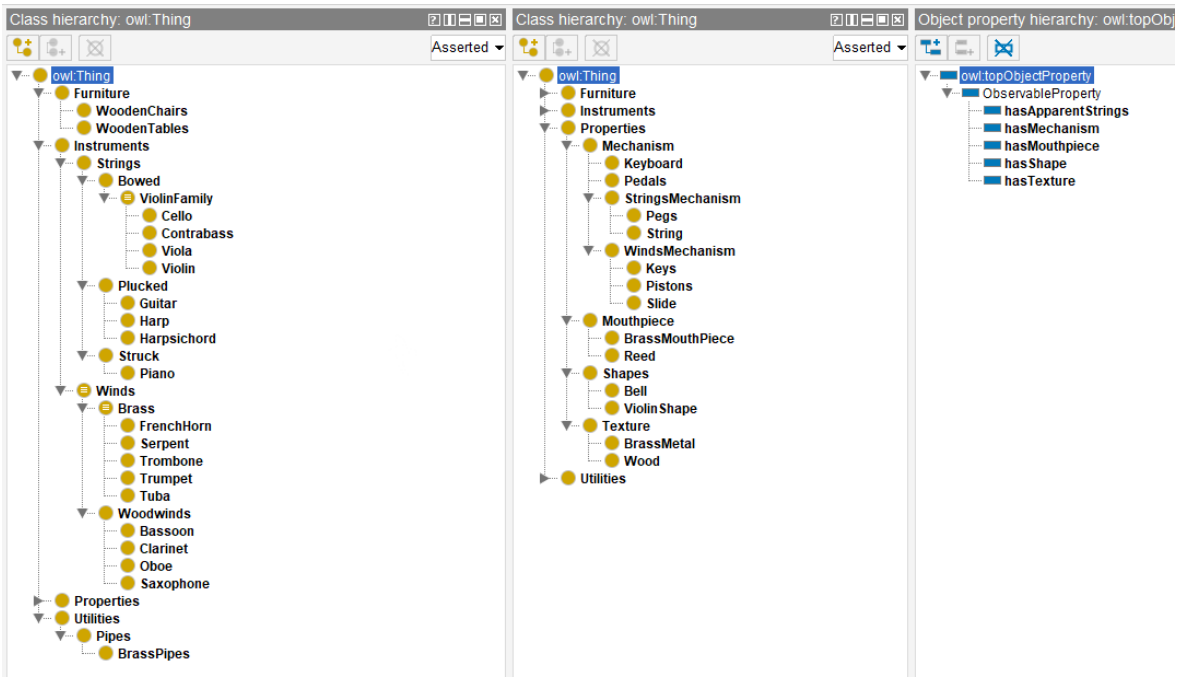


Figure 7.1: Classes and object properties hierarchies of the musical instruments ontology. From left to right, classes hierarchy of the ontology, classes hierarchy of the ranges of the object properties, object properties hierarchy of the ontology.

We identified five observable characteristics of musical instruments: their texture (e.g. wood or brass), their shape (e.g. a bell for most wind instruments or an hourglass shape for violins or guitars), their mechanism (e.g. keys, strings, slide, pistons), the presence of apparent strings (e.g. violins have apparent strings, pianos do not have apparent strings) and the type of mouthpiece for wind instruments (e.g. reeds or brass mouthpieces). We selected musical instruments that have overlapping combinations of observable characteristics. In order to evaluate the generalization power of the models built with OBIC, we introduced other classes such as wooden furniture and brass pipes. As a result, there are 17 musical instruments and 3 miscellaneous classes as the classes of our dataset and ontology. The classes are shown in Figure 7.1

7.1.1 The musical instruments ontology

We have used Protégé [144] to edit the ontologies. Figure 7.1 shows the class hierarchies and object properties of the ontology. The main class `Instruments` contains every instrument definition. We note that the musical instruments hierarchy is based on the Hornbostel-Sachs musical instruments classification [223]. This classification is not necessarily linked to observable characteristics of the instruments. The additional classes `Furniture` and `Utilities` represent miscellaneous classes that will be used to test the capacity of OBIC to handle different objects that share some characteristics. The five observable characteristics were defined as sub-properties of `:ObservableProperty`. The class `Properties` corresponds to all the ranges of the observable characteristics.

`:hasApparentStrings` This property describes whether some strings are apparent on the instrument. Its domain is the strings instruments represented by the class `Strings` and its range is the class `String`. This property was not defined as a `DataProperty` since it could be later used to determine the type of strings if subclasses of the mechanism `String` are defined. The class `ViolinFamily` is defined as having apparent strings, so are the classes `Guitar` and `Harp`.

`:hasMouthpiece` This property is analogous to the `:hasApparentStrings` property. It describes

whether a mouthpiece is apparent on the instrument. Only wind instruments have this property thus its domain is `Winds` and its range is `Mouthpiece`. We define the class `Winds` as equivalent to having exactly one mouthpiece[1]. It is a functional property because an instrument can only have one type of mouthpiece. Brass instruments are defined as having a brass mouthpiece and woodwinds are defined as having a mouthpiece that is not a brass mouthpiece. Indeed, there are several types of mouthpieces for woodwinds, the main one being reeds.

`:hasMechanism` This property depicts the type of mechanism that is visible on a musical instrument e.g. strings or keyboard. Its domain is `Instruments` and its range is `Mechanism`. Instruments can have several mechanisms. For instance, a violin has strings to play and pegs to tune it. Some mechanisms are specific to families of instruments. Subclasses of `WindsMechansim` and `StringsMechanism` are mechanisms that only wind instruments and strings instrument can have, respectively. Despite the fact that every instrument has some sort of mechanism to produce sound, the class `Instruments` does not have this property in its definition as it only concerns observable mechanisms. For example, the vocal chords are the mechanism that produces the human voice but they are never visible on images.

`:hasShape` This property refers to a particular shape that can be observed on an object. Its domain is `owl.Thing` and its range is `Shapes`, because any object can have a shape. We identified two main shapes that instruments have: a bell shape and a violin shape. The bell shape is observed on any instrument that has a bell e.g. wind instruments such as a saxophone or a trumpet. The violin shape describes the signature hourglass shape that many strings instruments have e.g. cello, guitar or violin.

`:hasTexture` This property is similar to the `:hasMaterial` property defined in Example 5.1 on page 85. It illustrates the type of texture of an instrument that is correlated to the type of material. Its domain is `owl.Thing` since any object has a texture, its range is `Texture`. Two main textures are attributed to musical instruments: wood and brass. To avoid ambiguities with the brass family, the texture brass is called `BrassMetal`. Nevertheless, an instrument of the brass family is not necessarily made of brass metal and vice-versa, a woodwind instrument is not necessarily made of wood. For instance, a saxophone is mostly made of brass metal while a serpent (an ancestor of the tuba) is made of wood and leather.

Several classes are defined as equivalent to a set of restrictions. We have already mentioned the definitions of `Winds` and `Brass` in the discussion of the property `:hasMouthpiece`. The class `ViolinFamily` is defined as having apparent strings, `Pegs` and `String` as mechanism, the shape `ViolinShape` and the texture `Wood`. The other classes are subclasses of anonymous classes. For instance, the class `Trombone` is a subclass of `Brass`, that has the mechanism `Slide` and not `Keys`, the texture `BrassMetal` and not `Wood` and the shape `Bell`. Other restrictions are inherited from the class `Brass`. The class definitions of every class of the ontology are available in Appendix B.

7.1.2 The musical instruments dataset

A dataset of images that correspond to the classes of the ontology was semi-automatically created. Images were gathered by automatically scraping Google Images for open-source pictures with the name of each class with the addition of the keywords *musical instrument*. The maximum number of images for one class was set to 500 although no class ended up with this many instances. Afterwards, we roughly inspected the resulting images to remove irrelevant images (e.g. the results for trombone had some images of paper clips that had to be removed because the two words are homonyms in French). This process was applied to demonstrate that the OBIC framework can easily be applied when a practitioner only has a dataset or an ontology available. A small and simple ontology that only describes the data can be built with little effort. Likewise, a dataset can be created semi-automatically with minimal manual work as we showed in this section.

[1] There are some exceptions to this rule in the Hornbostel-Sachs classification [223] that we ignore here for simplicity.

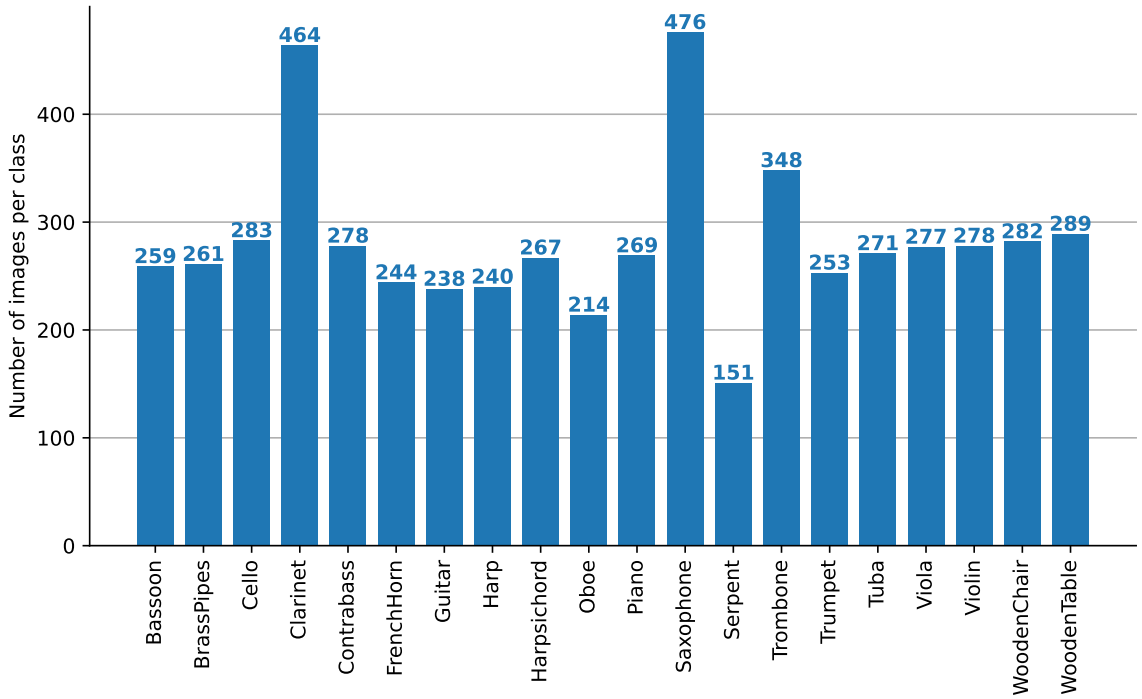


Figure 7.2: Class distribution in the musical instruments dataset

The resulting dataset contains 5642 images for 20 classes. Figure 7.2 shows the class distribution of the dataset. There are 282 images per class on average. The classes `Saxophone` and `Clarinet` are over-represented with 464 and 476 images respectively. Conversely, the class `Serpent` is under-represented with 151 images. The images were resized to a square resolution of 224×224 to make them compatible with some pre-trained models.

7.2 Experiments on OBIC

We want to evaluate the capacity of OBIC to accurately tell when a prediction is correct or incorrect on the described musical instrument classification task. Specifically, we measure the ability of the error detection system to tell when the *global classifier* was wrong. If OBIC does not significantly decrease the performance of the *global classifier* when rejecting detected errors, then this framework is a net gain in explainability and reliability as we would have improved the ability of the system to explain itself and behave in an expected manner without decreasing the classifier's performance.

7.2.1 Methodology

The first step of this evaluation is to build classifiers according to the process discussed in Chapter 5. The goal is to evaluate the ability of OBIC to detect errors made by the *global classifier*. An error detection system is a binary classifier that predicts the presence or absence of an error. As such, the positive class represents the presence of an error and the negative class represents the absence of an error. Hence, four types of results are possible: true positive, true negative, false positive and false negative. In this experiment, an error occurs when the *global classifier* does not predict the correct class. The error detection system is the ontology's consistency. An error is detected when the ontology is inconsistent. The interpretation of the four types of results in the context of OBIC is the following:

True Negative (TN) The *global classifier* predicted the correct class and the individual is consistent. There is no error and the error detection did not detect an error.

True Positive (TP) The *global classifier* did not predict the correct class and the resulting individual is inconsistent.

False Positive (FP) The *global classifier* predicted the correct class but the resulting individual is inconsistent.

False Negative (FN) The *global classifier* did not predict the correct class but the individual is consistent.

The number of TP, TN, FP and FN depends on the decision threshold of the binary classifier. Indeed, a classifier commonly outputs a prediction or confidence score for each class which is a continuous value between 0 and 1. The resulting predicted class depends on the decision threshold e.g. a score below 0.5 leads to the prediction of the negative class, 0.5 being the decision threshold. Therefore, the threshold affects the performance metrics of a classifier. To avoid this issue, AUROC and AUPR metrics are threshold-independent metrics. They require the score of each example to determine the number of instances of each category at several decision thresholds. However, the consistency of an ontology is not a score but a boolean value so the AUROC and AUPR metrics are not computable for the error detection system of OBIC. A choice of threshold still exists in OBIC to decide whether to add an assertion to the individual. We will study the impact of the thresholds on the error detection performance, by studying classical metrics for binary classification (e.g. precision, recall, F1-score...) at different thresholds.

A 5-fold cross-validation is conducted to gather the results where each fold represents the original class distribution. The metrics are then calculated for each fold and aggregated using the average value. For each fold, 6 pairs of thresholds values are tested: $\{(0,1),(0.1,0.9),(0.2,0.8),(0.3,0.7),(0.4,0.6),(0.5,0.5)\}$. Figure 7.3 is an illustration of the functioning of the thresholds described in Section 5.2.3. When $threshold^+ = 1$, the only outputs that leads to an assertion are 0 and 1. Hence for this threshold, we expect that the individuals are always consistent meaning that no error is detected. When $threshold^+ = 0.5$, every output leads to either a positive or negative assertion, thus it is expected to be the threshold with the most inconsistencies.

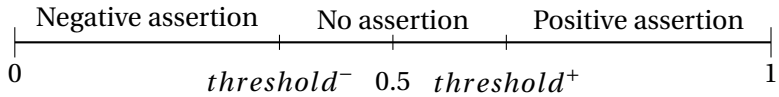


Figure 7.3: Illustration of the thresholds in OBIC. A negative assertion is added to the individual when the corresponding output of the classifier is less than $threshold^-$. Likewise, a positive assertion is added when the output is greater than $threshold^+$. Otherwise, the presence of the assertion is uncertain and nothing is added.

7.2.2 OBIC implementation

We applied the OBIC framework to the musical instrument classification task. The dataset and ontology have already been discussed in Section 7.1. We have compared several popular architectures (i.e. Resnet50 [224], Alexnet [225], VGG [226], Densenet [227] and Inception v3 [228]) and found that the ResNet50 architecture [224] gave the highest F1-score and accuracy. Hence, the *global classifier* and the property classifiers are all convolutional neural networks following this architecture. These models were pretrained on the ImageNet dataset [133] and were finetuned with our dataset. The finetuning was done by adding a fully-connected layer corresponding to the output layer of the model. The images of the dataset were normalized and cropped to fit the input size of ResNet50. The *global classifier* and the classifier for the functional property `:hasMouthpiece` were finetuned with a categorical cross-entropy loss and the activation function for the last layer is the softmax function. The non-functional property classifiers were finetuned with a binary cross-entropy loss and the activation function for the last layer is the sigmoid function.

Classifier	Accuracy (SD)	F1-score (SD)	AUROC (SD)
Global	0.834 (0.062)	0.827 (0.0063)	0.990 (0.0010)
`:hasMouthpiece*`	0.960 (0.0046)	0.956 (0.0053)	0.996 (0.0009)
`:hasApparentStrings`	0.972 (0.0030)	0.972 (0.0030)	0.996 (0.0005)
`:hasMechanism`	0.455 (0.0056)	0.559 (0.0065)	0.803 (0.0044)
`:hasShape`	0.925 (0.0081)	0.933 (0.0072)	0.992 (0.0014)
`:hasTexture`	0.946 (0.0061)	0.956 (0.0055)	0.991 (0.0018)

Table 7.1: Average and Standard Deviation (SD) of each classifier's performance over the 5-fold cross-validation. The `hasMouthpiece` property is a functional property, indicated with a *. Each metric is between 0 and 1, higher is better.

We measure the models' performance using the accuracy, F1-score and AUROC metrics. The accuracy for non-functional properties is the proportion of predictions that exactly match the true label e.g. the output $(0,1,1)$ is exact if the true label is $(0,1,1)$, otherwise it is not considered exact. The dataset is slightly imbalanced so we choose metrics that are insensitive to this issue. Consequently, for functional models, AUROC is calculated with the one-vs-one strategy. The F1-score for non-functional models are computed with the "micro" averaging strategy while the "macro" averaging is used for functional models to avoid redundancy with the accuracy.

Table 7.1 presents the performance of each classifier averaged over the five folds of the cross-validation. The performance of the models is overall satisfying except for the `:hasMechanism` classifier which has difficulties to correctly identify the presence of every class. This can be explained by the number of possible classes and the lack of variety in the combination of classes (e.g. pegs and strings are always together for most string instruments). It prevents the model to correctly identify a mechanism and distinguish one mechanism from another. Adding more instruments with other combinations of mechanisms to the dataset and ontology may improve the performance of this classifier.

7.2.3 Results

In this section, we present the results of the error detection system. In these results, we observed that the `:hasMechanism` classifier has significantly lower accuracy and F1-scores compared to the other classifiers. Thus, we investigate the impact of this difference in performance by running the error detection system without the assertions made by this classifier. Finally, we discuss the choice of metric to determine the ideal threshold values.

Results with every classifier

For each fold of the cross-validation, the test dataset contains an average of 1128.4 images. The accuracy of the *global* classifier is 0.83 (see Table 7.1) meaning that there is an average of 187.6 classification errors per fold. The average number of instances of each category (i.e. true negatives, true positives, false positives and false negatives) for each threshold are shown in Figure 7.4. The performance of the error detection system is clearer in Figure 7.5 where the average precision, recall, F1-score and False Positive Rate (FPR) are shown. We observe that the precision follows an increasing trend, contrary to the other metrics. However, the difference between the values of each fold also increases, with a significant difference between minimum and maximum precision with $threshold^{+} = 0.9$ This is due to the fact that for high threshold values, few instances are inconsistent which renders the precision metric unstable. The instability is not reflected in the F1-score because the harmonic mean tends towards the smallest elements, in this case the recall.

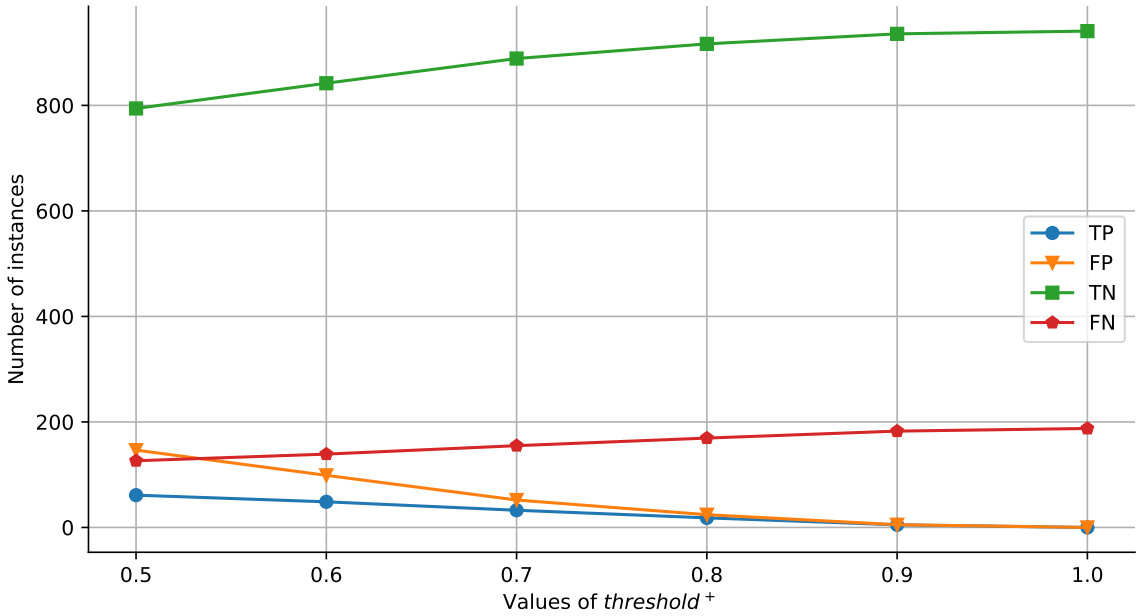


Figure 7.4: Average number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN) for each threshold. The number of predicted errors (i.e. FP and TP) is highest for $threshold^+ = 0.5$ and decreases until it reaches 0 when $threshold^+ = 1$, as expected. The number of true positives is always lower than the number of false positives for every threshold except for 0.9 where it balances out.

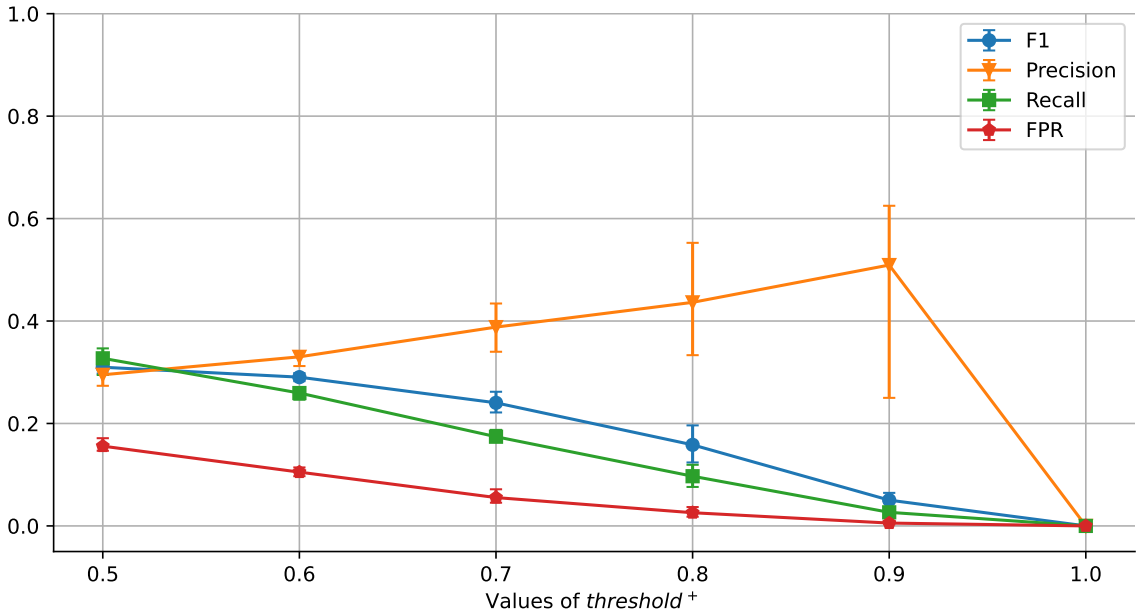


Figure 7.5: Average precision, recall, F1-score and FPR for each threshold. Error bars represent the minimum and maximum observed values on the 5 folds. All metrics are between 0 and 1. For precision, recall and F1-score, higher is better. Inversely, lower is better for the FPR. The F1-score, recall and FPR all follow the same decreasing trend; their highest value is reached with $threshold^+ = 0.5$ and lowest with $threshold^+ = 1$. Regarding precision, it follows an increasing trend for threshold values of 0.5 to 0.9 and falls down to 0 with $threshold^+ = 1$ because there is no positive prediction at this value. We note that the extreme values of precision become increasingly distant when $threshold^+$ increases.

Results without the `:hasMechanism` classifier

We hypothesized that the performance of the error detection may negatively affect by the `:hasMechanism` classifier, that showed worse performance than the other classifiers. In order to measure this effect, we ran the same experiments but removed the assertions produced by the `:hasMechanism` classifier. The impact of removing `:hasMechanism` from the assertions on the number of TP, FP, FN and FP is presented in Table 7.2. Figure 7.6 shows the evolution of each score when removing the predictions of `:hasMechanism`.

$threshold^+$	TP	FP	TN	FN
0.5	-24.5 %	-51.6 %	9.5 %	11.9 %
0.6	-31.7 %	-57.3 %	6.7 %	11.1 %
0.7	-36.2 %	-57.7 %	3.3 %	8.3 %
0.8	-48.4 %	-69.4 %	1.8 %	1.5 %
0.9	-48.0 %	-80.8 %	0.4 %	0.2 %

Table 7.2: Evolution of the quantity of prediction in each category when removing assertions from the `:hasMechanism` classifier. For instance, with $threshold^+ = 0.5$, the error detection system without `:hasMechanism` assertions has predicted 24.5% less TP than the same system with the assertions from every classifier. Ideally, the number of TP and TN has increased (i.e. a positive evolution) and the number of FP and FN has decreased (i.e. a negative evolution).
The number of positive predictions (TP and FP) has decreased but we observe that FP have decreased more than TP. Consequently, negative predictions (TN and FN) have increased, with a larger increase in TN than FN.

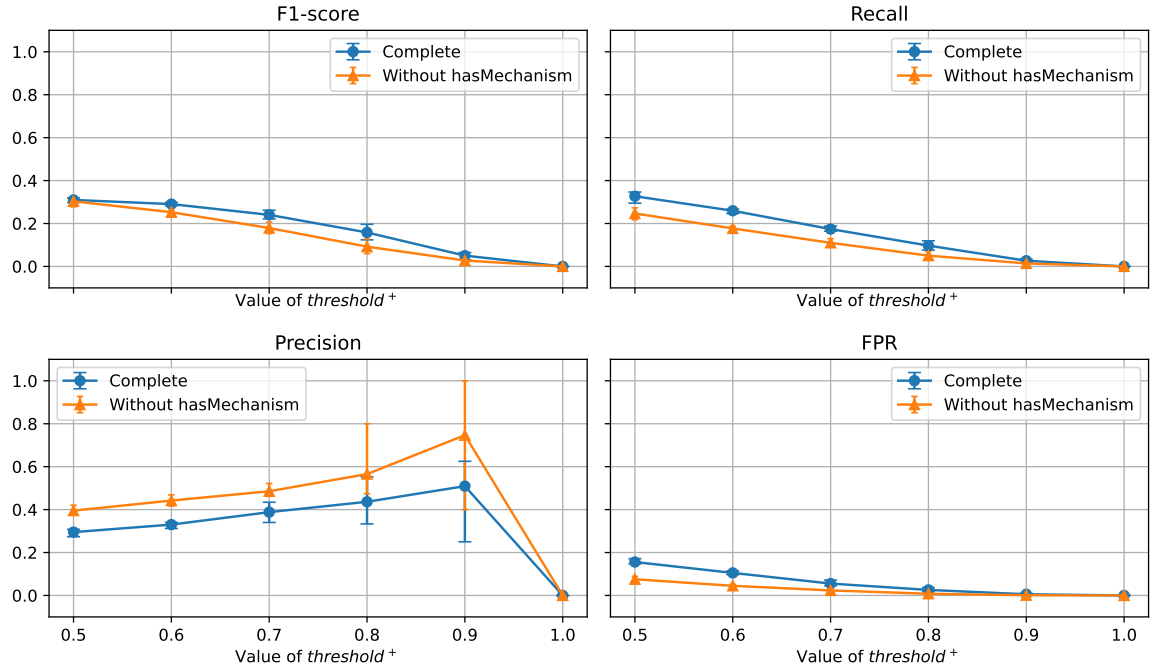


Figure 7.6: Evolution of classification scores when removing `:hasMechanism` classifier. For F1-score, recall and precision, higher is better. For FPR, lower is better.
Despite the notable decrease in FP, the F1-score and recall have deteriorated due to the decrease in TP and increase in FN. Conversely, precision and FPR have improved. The precision is still affected by the instability provoked by the low number of TP and FP for high thresholds (e.g. 0.8 and 0.9).

Overall, there is a clear decrease in the number of inconsistencies which leads to less TP and FP.

Most inconsistencies provoked by the `:hasMechanism` classifier were FP i.e. the *global classifier* predicted the correct class but the individual was inconsistent. As a result of this large decrease in FP, the precision score and FPR were improved compared to the original results. However, as a consequence of this decrease of inconsistencies, the number of FN increased. False negatives were already predominant before removing the `:hasMechanism` assertions which resulted in a low recall and F1-score. Therefore, this removal worsened the recall and consequently the F1-score.

Determining the ideal thresholds

This experiment was conducted to observe the behavior of the error detection system of OBIC according to the threshold. It is clear that the threshold value has a large influence on the results. To determine the best threshold, we argue that we should look for the threshold value that maximizes precision and minimizes FPR. Indeed, precision represents the proportion of true positives among all the positive predictions while FPR represents the probability of raising a false alarm. Our goal with this error detection system is to minimize the number of false alarms while maximizing the number of correct predictions. Hence, we propose to plot the FPR-Precision curve where one point corresponds to the FPR for the x-axis and precision score for the y-axis of one threshold. We look for the point of the curve that is the closest to the point $(0,1)$ as it corresponds to having no false positive predictions. Figure 7.7 presents these curves for the experiments with and without the predictions from the `:hasMechanism` classifier. This graph shows the improvement on precision and FPR achieved by removing the assertions from `:hasMechanism`. It seems that the best threshold value is 0.9 as it has the best pair of precision and FPR values. Nevertheless, we noted that the precision score is unstable and the values vary significantly depending on the cross-validation fold. The threshold value of 0.8 is more conservative and achieves better precision in the worst case.

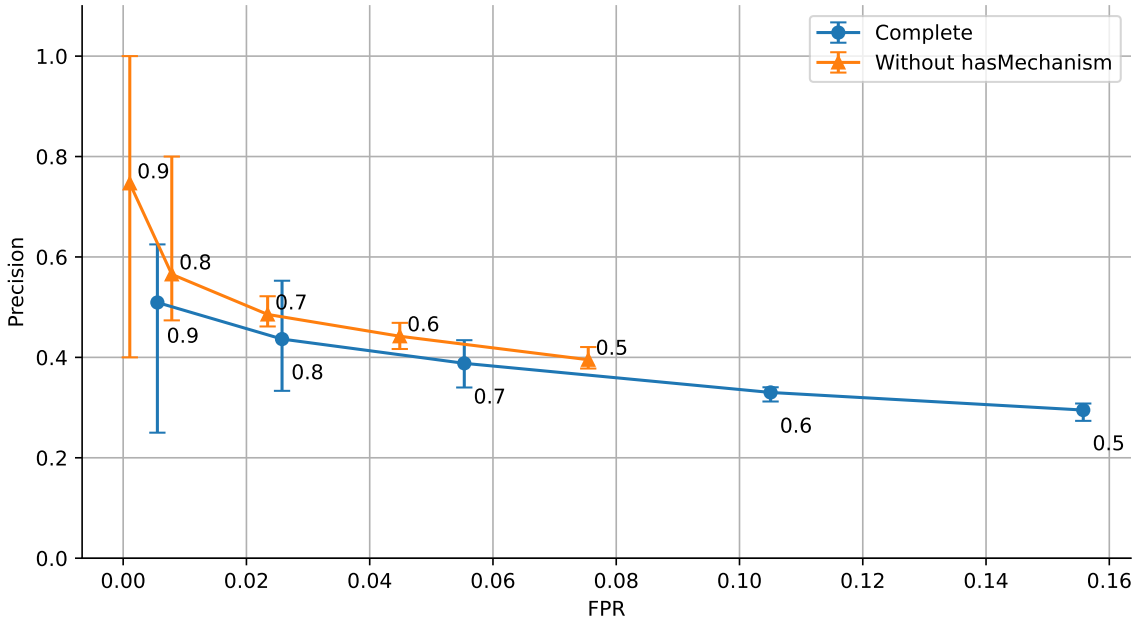


Figure 7.7: FPR-Precision curve with and without `:hasMechanism` classifier

7.2.4 Analysis

The error detection system within OBIC detects an error when there is an inconsistency in the individual. An inconsistency is always caused by a wrong prediction in at least one model (either property classifier or the *global classifier*). Hence, the error detection system is able to detect when any wrong prediction asserted to the individual but is not able to tell which model made the

mistake. In this experiment, we measure and analyze the ability to detect the errors made by the *global classifier*. Any error detected by the system is attributed to the *global classifier* as it is the standard behavior of error detection methods. The results show that the error detection does not catch most errors as there is a high proportion of false negatives. Furthermore, the number of false positives is greater than the number of true positives except for threshold values greater than 0.8, where the number of detected errors decreases significantly. Indeed, the *global classifier* made 187 errors in total but at these threshold values, the number of predicted errors is in the magnitude of 10% of the actual number of errors.

In this section, we analyze the impact of removing the `:hasMechanism` assertions. The effect of this particular classifier gives an overview of how the performance of a property classifier influences the error detection. Then, we study the causes of false negative and false positive predictions. Finally, we discuss the effect of the threshold values on the error detection performance and how to select the best threshold value.

Impact of removing the `:hasMechanism` assertions

We formulated and tested the hypothesis that the weak performance of the `:hasMechanism` classifier was the source of most false positive predictions. Removing the `:hasMechanism` model decreased the number of false positives by at least 50%. However, it also reduced the total number of positive predictions and thus did not improve the quality of the error detection. The object property `:hasMechanism` plays an important part in detecting the inconsistency of an individual in the musical instrument ontology. Indeed, the range of this property contains many different classes that differentiate instruments with similar properties e.g. the only difference among a trumpet, a trombone or a horn is their mechanism. Losing the information about the mechanism induces a decrease in the number of inconsistencies and led to the observed consequences on the error detection. We attribute the poor performance of the model to the large number of classes to detect and the fact that some mechanisms always appear together. The `:hasMechanism` classifier predicts 7 different classes which are imbalanced in the dataset since some mechanisms are unique to an instrument in the ontology. Moreover, some mechanisms such as the pegs and strings are always paired together which prevents the model to distinguish these two mechanisms.

This shows that class definitions should not rely on one observable property to be differentiated from other classes. Likewise, correlations between objects of a property should be avoided e.g. pegs and strings are strongly correlated which may negatively impact the classifier's ability to distinguish them. We have confirmed that the performance of a property classifier may significantly impact the performance of the error detection. Hence, appropriate measures must be taken to maximize a classifier's performance, such as finding the right balance between the amount of training data and the number of classes to detect and use different architectures for each classifier to better suit their task and data.

Causes of false positives

False positives are the consequence of an error made by at least one property classifier. The functioning of OBIC relies on building as many models as there are observable properties. Yet, the more property classifiers are used, the more failure points are introduced in the system. In this experiment, there are more false positives than true positives because it is more likely that one property classifier made a mistake. The accuracy of the *global classifier* is 0.834 meaning that there is 17% chance that it makes an error. Regarding the property classifier function as a whole, the chances that at least one property classifier makes a wrong prediction is the sum of the probability that one model makes an error. This probability is 20% without the model for `:hasMechanism` and rises up to 74% when including this model. The thresholds and construction of the ontology lessen this effect as not every prediction from these models is added to the individual and some wrong predictions may not provoke inconsistencies. Still, this shows that the property classifiers are more likely to fail than the *global classifier*, leading to a greater number of false positives than true pos-

itives. We expect that this phenomenon inverses when the *global classifier* performs worse than the sum of the property classifiers. Although deteriorating the performance of the *global classifier* would result in seemingly better performance of the error detection, it is not desirable as the main objective is to minimize the number of errors. Solutions to reduce the number of FP are to improve the property classifiers performance and reduce the number of property classifiers.

Another source of false positives is the presence of other objects in the image that result in inconsistent predictions from the property classifiers. For instance, parts of another instrument may be present in an image. In an ideal case where the property classifiers function perfectly, they would detect properties of this other instrument. It is assumed that there is only one instrument in the image, thus an inconsistency would occur even though all the classifiers were correct. This situation is rare and cannot be addressed in the error detection system. Nevertheless, humans can observe and understand this issue thanks to the explanation interface.

Causes of false negatives

False negatives are errors from the *global classifier* that did not lead to an inconsistency. An inconsistency occurs when the predictions from the property classifiers are not in agreement with the predicted class. A false negative can be explained by several concurrent causes.

- The first cause may be that the property classifiers made wrong predictions that are consistent with the incorrect predicted class. In this experiment, the property classifiers and the *global classifier* all stem from the same architecture with the same initial weights, that were then finetuned with the same images but with different labels. In addition, some properties are highly correlated with the main class e.g. only trombones have slide in the dataset. Therefore, when the *global classifier* mistakenly predicts a trombone, it is likely that the property classifiers also mistakenly detected properties specific to a trombone.

- Another source of false negatives is the ontology design that may not be restrictive enough to raise inconsistencies in the presence of some assertions. For instance, it is not explicit in the ontology that a piano does not have a brass texture because some elements of a piano may look like brass (e.g. the wheels or internal parts are similar to the brass texture). Thus the detection of a brass texture by a property classifier is consistent with a piano even though the `:hasTexture` classifier or the *global classifier* made a wrong prediction.

- Finally, a cause of false negative predictions is that some assertions are not added because the corresponding prediction score is between the two thresholds. Some of these assertions may have provoked an inconsistency and thus an error in the main class would have been detected. This issue is related to the problem of choosing an adequate threshold.

Effect and choice of the thresholds

The thresholds are used to handle uncertain predictions, a low value of $threshold^+$ (e.g. 0.5) is chosen when the property classifiers have exceptional performance and are reliable. Inversely, a high value of $threshold^+$ (e.g. 0.9) means that the model does not perform well and only the most certain predictions are added as assertions. The thresholds help lower the impact of wrong predictions made by the property classifiers. High values are more conservative but cause a higher number of false negatives compared to low values. Our observations of the impact of the `:hasMechanism` classifier on the error detection lead us to argue that each property classifier should have their own thresholds, chosen according to the model's performance. However, determining the threshold values is not trivial and having to make such a choice for every model would severely increase the difficulty to use OBIC. Another direction to improve the impact of the threshold is to use a different confidence score. We have discussed in Section 5.1.1 that the output scores of machine learning classifiers should not be trusted because they tend to always have a high confidence. Using alternative confidence scores such as the *trust* score [157] may simplify the choice of thresholds because the confidence scores would be more reliable.

Regarding the choice of threshold, we proposed to identify thresholds that minimize the false positives while maximizing true positives. We selected the false positive rate and the precision as metrics to observe when determining the ideal threshold. The FPR can be interpreted as the probability to raise a false alarm i.e. mistakenly detect an error. Precision is the proportion of correct positive predictions among all positive predictions. Hence, we seek the threshold values where the FPR is minimal and the precision is maximal. However, these metrics are not proportional to the number of detected errors. We have seen that high threshold values detect a small portion of the errors which skews precision and FPR. The design of a metric that encompasses both FPR and precision and that is weighted by the number of positive predictions may enable the automation of the choice of thresholds. With this metric, choosing the thresholds would become similar to finding the best hyperparameters for a machine learning model with standard search methods (e.g. grid search, random search). Nevertheless, the design of such metric depends on the task and domain of application. Designers of sensitive applications may prefer to have many false alarms in order to avoid missing errors that would have a detrimental impact. In this case, the proposed metric is not suitable and a new metric that minimizes the false negative rate and maximizes the negative predictive value[2] (NPV) may be better suited.

Discussion

The main issue with the proposed evaluation approach is that we attributed every error to the *global classifier*. The functioning of OBIC means that an inconsistency is necessarily the outcome of an error in at least one classifier. The experiment showed that most inconsistencies are provoked by errors in the property classifiers. Indeed, false positives are more likely to occur when property classifiers perform worse than the *global classifier*. Furthermore, we observed that the quantity of property classifiers also govern the quality of the error detection as the probabilities that one classifier is wrong is proportional to the number of property classifiers. Hence, this evaluation shows that the unsatisfactory performance of OBIC are partly due to the fact that all errors come from the *global classifier*, which results in more false positives than true positives. Since we have the assurance that an inconsistency necessarily means that an error was made by one or multiple classifiers, the challenge is to determine the wrong classifiers. Thus, an additional step to determines the source of the error is required. The use of counterfactual explanations that seek the most simple changes to remove the inconsistency may be an interesting direction to determine the source of the error. Moreover, the utilization of better confidence score may also help to identify the wrong classifiers.

OBIC is able to detect inconsistencies in the predictions of the classifiers. This error detection system does not require to know the true class to detect the errors. To our knowledge, there are little work about such unsupervised error detection system. We discussed that the detection of an error implies that there is an inconsistency provoked by at least one classifier. Consequently, this error detection system is a net gain in explainability and reliability as it shares the same performance as a machine learning model while being capable of detecting some errors which cannot be done in traditional architectures. Despite the issues highlighted by this experiment, the main objectives of OBIC are fulfilled and we have identified directions to address the identified problems for future applications.

7.3 Experiments on the counterfactual explanations

We have conducted an evaluation of the objective quality of the counterfactual explanations generated by CEO in Chapter 6. We analyzed that the proximity metric applied to rank the counterfactuals favored abstract explanations, making them less relevant in our opinion. Nevertheless, scholars recommend caution about using intuition to assess the subjective quality of an explanation. Hence, we conducted a user-study based on the XIS discussed in Chapter 5 to achieve two

[2]The Negative Predictive Value is computed with the formula: $\frac{TN}{TN+FN}$.

goals. The first goal is to evaluate the relevance of counterfactual explanations to explain OBIC. The second goal is the quantification of the counterfactuals quality which is connected to the ranking of the counterfactuals and therefore the proximity metric. This study is a test of satisfaction which measures the participants' self-reported satisfaction, as described in [109].

Similar to the experiments on OBIC, we evaluate CEO on the musical instruments classification task described in Section 7.1. Specifically, the counterfactuals are about the predictions from OBIC. In the conclusion of Chapter 5, we expected the user to inquire explanations as to why an error was detected and what should be changed to make the ontology consistent. For instance, consider that a harpsichord is detected as well as pedals and a wooden texture; this is not consistent with the ontology because a harpsichord does not have pedals. The goal of the CEO method is to propose modifications for these inconsistent assertions so that the user understands which properties were wrong and how to change them. We display the input image and the main results of OBIC (it corresponds to the top sections of the XUI, see Section 5.3) along with the counterfactual explanations generated by CEO. Thus, this user-study allows us to test the reception of the XIS by the users as well as the quality of the counterfactuals. In this experiment, we do not evaluate the objective metrics introduced in Section 6.1.3 for several reasons. First, a goal of the conducted study is to improve the current proximity metric which will directly impact the objective metrics. Secondly, the objective metrics allow counterfactual explanation methods to be compared. Yet, the comparison was possible because the methods could be applied on similar machine learning tasks. To our knowledge, CEO is the first method to generate counterfactuals for ontologies. Therefore, it is not possible to apply CEO on the same tasks as the other methods. Moreover, the metrics would also have different meanings that prevent the comparison of CEO with the other methods.

7.3.1 Methodology

A user-study with domain experts is conducted to evaluate the CEO method. We were confronted to the recruitment issue to find participants discussed by Chromik and Schuessler [98]. Consequently, the survey was conducted on a sample of six domain experts i.e. experienced musicians. To compensate for this small sample size, we interviewed each participant after the survey to get their feedback and help the analysis of the results. The survey presented nine images and their corresponding classification and explanation with OBIC. Counterfactual explanations were generated by CEO and ranked by proximity and sparsity as mentioned in Chapter 6. The counterfactual question was stated (e.g. "What should change to classify a piano instead of a harpsichord ?") and the 10 top ranking counterfactuals were shown to the user in their ranking order. After each case, the users were asked three questions:

1. Did these explanations allow you to understand the required modifications to predict the other class ?
2. Which counterfactual explanation did you prefer ?
3. What explanations did you find relevant ?

The first question was answered on a scale from 1 to 4 where 1 corresponds to "Not at all" and 4 corresponds to "Yes, absolutely". It enables us to measure the relevance of the counterfactuals to answer the stated question. The second question required the users to select the one explanation they preferred. They also had the choice to say that they did not have any preference. This question gives information about the desired level of abstraction of the counterfactual explanations. Finally, the last question necessitated the user to tick the explanations that seemed relevant to them. Ideally, the ticked explanations are the top ranked explanations. This enables us to further evaluate the proximity metric and its ability to select the best explanations. At the end of the survey, the participants were asked whether they think this type of explanation is useful.

The nine cases composing the survey are nine different images of musical instruments extracted from the dataset. Particularly, the cases misclassified by the *global classifier* from OBIC

What did the AI algorithm detect ?

The AI algorithm determined that the instrument in the image is a **Harpsichord**.
It detected the following properties:

- Wood texture
- Brass texture
- A keyboard mechanism
- Pedals mechanism

An inconsistency is detected, these properties do not match a harpsichord.
The system proposes different explanations that answer the question "What changes on the properties detected should be made to make them consistent with a harpsichord ?"

1. **Replace brass texture with another texture** AND **replace pedals mechanism with another mechanism**.
2. **Replace brass texture with wood** AND replace pedals mechanism with another mechanism.
3. **Replace brass texture with another texture** AND **replace pedals mechanism with another string instrument mechanism**.
4. Replace brass texture with another texture AND **replace pedals mechanism with a keyboard**.

Figure 7.8: Description and the first four explanations of the first case of the survey

Figure 7.9: Input image of the first case of the survey.

were retained. Then, the counterfactual question depends on whether OBIC detected the error. The counterfactual question for the cases where the error was detected is: "What changes on the properties detected should be made to make them consistent with a <*predicted class*> ?" where <*predicted class*> is the name of the class that was predicted. The counterfactuals are expected to modify the identified faulty assertions by either deleting them or changing the class of their objects. The counterfactual question for the cases where the error was not detected is: ""What changes on the properties detected should be made to make them consistent with a <*true class*>?" where <*true class*> is replaced with the name of the correct class. Domain experts are able to correctly identify the correct class in the image. This counterfactual question will enable them to understand what went wrong in the system. The presentation format of a case and its counterfactual explanations are presented in Figure 7.8 along with the input image in Figure 7.9. The counterfactuals are presented as a text telling the user the changes to make, in no particular order.

For each case, we study the ranking of the preferred explanation, the proportion of relevant explanations and the quality of the ranking. The quality of the ranking is done by evaluating the ability of the ranking to place relevant explanations at the top ranks. The AUROC score is used in information retrieval to measure the capacity of a system to rank the relevance of documents [229]. Therefore, we calculate the AUROC score to evaluate the quality of the ranking of explanations which acts as proxy for the quality of the proximity metric. In this context, we plot the ROC curve as the number of relevant explanations on the y-axis and the number of non relevant explanations on the x-axis. An AUROC score is calculated for each expert, then the average of these scores is used to obtain the AUROC score of each case. We further evaluate the ranking by studying key statistics for each rank. Namely, the number of times explanations at each ranking were considered relevant by a majority of experts, the number of times a ranking was preferred by experts over every case and the number of times a ranking was consensually preferred. A consensus or majority is reached when the proportion of experts that agree on the relevance or preferred explanation is greater or equal to 50%. This study of the ranking will help us determine the level of abstraction that should be favored and thus choose a more appropriate proximity metric.

7.3.2 Results

The results of the study are combined in Tables 7.3 and 7.4. We observe in Table 7.3 that the number of relevant explanations is highly dependent on the case. Most cases have more than 40% of relevant explanations with the second case reaching 90%. Cases 3, 5 and 9 have the lowest proportion of relevant explanations with roughly 20% of relevant explanations. The common denom-

Case	Preferred explanations (% who chose it)	% of relevant explanations	Average AUROC
1	2nd (16.7%), 3rd (16.7%), **7th** (50%), 8th (16.7%)	70 %	0.77
2	5th (16.7%), **10th** (83.3%)	90%	0.55
3	1st (33.3%), **3rd** (50%), None (16.7%)	22%	0.93
4	6th (33.3%), 7th (16.7 %), 8th (33.3%), 10th (16.7%)	80%	0.55
5	**2nd** (83.3%), None (16.7 %)	20%	0.96
6	3rd (16.7%), 8th (16.7%), **9th** (50%), None (16.7%)	50%	0.76
7	1st (16.7%), **3rd** (50%), 9th (16.7%), None (16.7%)	40%	0.87
8	7th (33.3%), **8th** (66.7%)	80%	0.65
9	1st (16.7%), **3rd** (66.7%), 9th (16.7%)	20%	0.78

Table 7.3: Results of the user study for each case presented to six domain experts. Preferred explanations show which explanation were picked as the best explanation by the experts for each case. The percentage of relevant explanations goes from 0% to 100%, 100% being the ideal value which would mean that all presented explanations were relevant. The average AUROC goes from 0 to 1, higher is better. It presents the quality of the ranking i.e. whether relevant explanations are sorted in the top ranks while non relevant are at the bottom ranks.

Explanation ranking	Proportion of relevance	Preferred	Consensually preferred
1st	9/9 (100%)	4	0/9 (0%)
2nd	7/9 (78%)	6	1/9 (11%)
3rd	7/9 (78%)	12	3/9 (33%)
4th	4/9 (44%)	0	0/9 (0%)
5th	3/9 (33%)	1	0/9 (0%)
6th	4/9 (44%)	2	0/9 (0%)
7th	4/9 (44%)	6	1/9 (11%)
8th	4/9 (44%)	8	1/9 (11%)
9th	3/9 (33%)	5	1/9 (11%)
10th	2/9 (22%)	6	1/9 (11%)

Table 7.4: Results of the user study for each rank of explanation. It describes the quality of the ranking by showing the number of times the n-th ranked explanation was: relevant (higher is better), preferred by any expert and preferred by a majority of experts. It is expected that top ranked explanations are more relevant and more often preferred than the other explanations.

inator of these three cases is the inclusion of an assertion about the shape of the instrument. The participants identified a flaw in the ontology design concerning the `:hasShape` property thanks to the counterfactual explanations. The poor relevance of these cases is hence attributed to this issue. We removed these cases from Table 7.4 to create a table similar to Table 7.5 that is not impacted by the ontology design issue.

Table 7.4 clearly demonstrates that the top ranked explanation is always relevant, even in the cases where the counterfactuals were flawed because of the ontology design issue. Moreover, there is always more than one relevant explanation proposed, implying that the ranking guarantees some relevant explanations. The AUROC score is always greater than 0.5, indicating that the relevant explanations are not packed at the bottom of the ranking. This is confirmed by the proportion of relevance which steadily decreases according to the ranking of the explanation. Despite the guaranteed relevance of the top ranked counterfactuals, they are the least preferred by the participants. Indeed, the last four explanations are consensually preferred 4 out of 6 times with 24 preferences over a total of 34. The top 5 explanations were preferred 8 times and only the third explanation was consensually preferred once (in Table 7.5). However, the preferred explanations per case exposed in Table 7.3 show that a minority of participants systematically preferred the first explanations.

Explanation ranking	Proportion of relevance	Preferred	Consensually preferred
1st	6/6 (100%)	1	0/6 (0%)
2nd	6/6 (100%)	1	0/6 (0%)
3rd	5/6 (83%)	5	1/6 (16.7%)
4th	4/6 (67%)	0	0/6 (0%)
5th	3/6 (50%)	1	0/6 (0%)
6th	4/6 (67%)	2	0/6 (0%)
7th	4/6 (67%)	6	1/6 (16.7%)
8th	4/6 (67%)	8	1/6 (16.7%)
9th	3/6 (50%)	4	1/6 (16.7%)
10th	2/6 (33%)	6	1/6 (16.7%)

Table 7.5: Results of the user study for each rank of explanation after removing cases 3, 5 and 9. This table is similar to Table 7.4.

We discussed in Chapter 6 that the proximity metric applied to rank the explanations leads to abstract explanations at the top ranks. This phenomenon can be observed in the content of the explanations in this experiment. The top ranked explanations successfully identify the faulty assertions but give a generic change e.g. "replace wood by another texture". As a result, the preferred explanations are never at the top ranks because experts sought more specific changes that have the same level of abstraction as the original assertion. Hence, the preferred explanations are generally at the bottom of the ranking. In addition, there are groups of explanations that have the same proximity and sparsity while exploring different changes. This happens when there are multiple possible classes at the same level of abstraction. For instance, in the first case, the 3rd to 6th explanations explore modifications on the mechanism without modifying any other assertions. This limits the diversity of the explanations as these groups carry the same idea without exploring alternatives that would increase the diversity and consequently increase the quality of the explanations. Finally, as expected, the counterfactuals that include a deletion operation are positioned at the bottom of the ranking in accordance to the design of the proximity metric.

We interviewed the participants after the survey to get their feedback and their feelings about the survey and the explanations. Overall, they complained that the interpretation of the explanations necessitated an important cognitive effort. They said that there was too many explanations and their presentation was problematic. They suggested an interactive interface rather than using static text. They deplored the absence of insertion operations that could have been highly relevant in some cases. For instance, wind instruments always have a mouthpiece, yet when a mouthpiece was not present in the original prediction, it was not added to the counterfactuals though experts expected the addition of a mouthpiece in the explanation. Despite these issues, they unanimously stated that counterfactual explanations were useful to understand the decision process from OBIC. Moreover, these explanations allowed them to point out a design flaw in the ontology without any prior knowledge of ontologies. It allowed them to understand the functioning and limitations of the predictive algorithm and helped them decide whether to trust a decision by the system.

7.3.3 Analysis

The first goal of this study was to evaluate the relevance of counterfactual explanations to explain the decision and error detection of the OBIC framework. The six interrogated domain experts said that these explanations were helpful to understand the decision process and the error detection system. Despite their apparent and unanimous satisfaction, they struggled to understand and interpret the explanations because they imposed a high cognitive load. Indeed, the cognitive effort is a major component for interpretability as discussed in Chapter 4. Additional work must be done on the presentation of the counterfactuals. The integration of CEO in an XUI with some

level of interactivity was identified by the participants as a way to decrease the required cognitive effort. The user would be free to choose the foil class and modify key elements of the CEO method such as the similarity metric. A graph representation of the counterfactuals could further help the user identify and visualize clusters of counterfactuals that modify the same assertion, remedying at the same time the problem of groups of similar counterfactuals identified in the results. Still, the conducted test of satisfaction is conclusive and the CEO method to explain OBIC is relevant, provided the problems raised in this experiment are solved.

The second objective of this study was to evaluate the ranking and proximity metric. The current ranking method only uses the proximity and sparsity metrics to assess the best and most relevant counterfactuals. The results show a good ability to place relevant explanations at the top of the ranking. This is reflected by the AUROC scores that are scattered between 0.5 and 0.9 indicating a satisfying ranking that mostly puts the relevant explanations at the top ranks. Yet, Table 7.3 shows that the preferred and therefore most relevant explanations are at the bottom of the ranking and never at the same position. This is an issue because the participants complained that there were too many explanations which negatively impacted their satisfaction. Considering that there is never 100% of relevant explanations, it seems possible to reduce the number of presented counterfactuals. However, it is clear that the ranking is not good enough to only present relevant explanations with a decreased number of presented counterfactuals. For instance, if 5 explanations were retained instead of 10, most of the preferred explanations would not be presented with the current ranking. Furthermore, it is possible that there are better explanations in the rest of the generated counterfactuals that are not in the top 10.

In Section 7.3.2, we attributed the lack of preferred explanations in the top ranks to the level of abstraction that is favored by the current proximity metric. Indeed, we observed that the changes presented in the counterfactuals with the lowest proximity identify the relevant assertions to modify but the modification is always one level of abstraction higher than the original class (e.g. change the texture `Wood` to the texture `Texture`). Despite our intuition and a majority of the preferred explanations by the participants, a minority of interrogated experts still preferred the first explanations. The limited sample size prevents us to make any further conclusion. We make the assumption that some users may prefer these abstract explanations. Hence, the diversity metric should be taken into account for the ranking of the counterfactuals so that explanations with different levels of abstraction are presented. In addition, this would avoid the groups of counterfactuals that are similar and share the same level of proximity.

Beyond the possible improvements discussed above, the proximity metric may be improved to better suit the task. The majority of experts preferred to be shown modifications that have the same depth as the original class (e.g. change the texture `Wood` to the texture `BrassMetal`). The current similarity metric considers that the neighboring classes depth-wise are more similar than classes at the same depth. It should be the other way around, classes on the same depth or level of abstraction should be considered more similar if they share a common ancestor. Besides, the proximity metric penalizes assertion deletion operations resulting in a low ranking for these explanations. However, the bottom explanations were mostly preferred which, in some cases, contained deletion operations. The deletion operation was indeed expected by the experts with this musical instruments ontology e.g. the deletion of a second mechanism in an instrument that only has one is expected instead of modifying the second mechanism into another one. The cost function for the deletion operation should be reworked in a way that is adequate to the task. Nevertheless, it becomes apparent that the proximity metric should be adapted to the task and the ontology. Finding an adequate proximity metric for each application is a challenge that needs to be addressed to efficiently apply the CEO method. The proposed choice of proximity may be sufficient when the diversity is incorporated to the ranking solution.

Finally, the participants pointed out known limitations of the CEO method i.e. the lack of insertion operations in the exploration and the confusing nature of the deletion operation. Indeed, some cases could have greatly benefited from insertion operations such as adding a mouthpiece to wind instruments when the mouthpiece was not detected. The exploration of the graph of pos-

sible counterfactuals must be reworked to include such operation efficiently. However, the size of the search space would be greatly increased because of the insertion operation which is not desirable considering the computation time required per individual. Regarding deletion operations, participants did not understand the meaning of this type of operation with the open-world assumption. They thought that deleting an assertion was equivalent to making explicit its absence, which is not the case with the open-world assumption. A way to render deletion operations more intuitive is to add `NegativeObjectPropertyAssertions` to the counterfactual IKG when a deletion operation is applied. Finally, `DataPropertyAssertions` are not supported by CEO, which hinders the capacity of CEO to be compatible with every ontology.

7.4 Conclusion

We have run experiments to evaluate the error detection system of OBIC as well as the relevance and quality of the explanations generated with CEO on the same task. The experiment on OBIC allowed us to fully implement and test this system on a specific task. The automatic building and training of the machine learning models fulfilled our expectations by making the building of the classifiers simple. The evaluation of the error detection system demonstrated that OBIC is capable of detecting errors through inconsistencies. However, the evaluation approach skewed the results because we focused on detecting errors from the *global classifier* only. It highlighted that OBIC lacks a way to identify the classifiers that provoked an inconsistency. We also gained information on the behavior of OBIC with regards to the design of the ontology. This information enables the design of methodologies to create adapted ontologies and automate the choice of thresholds. Furthermore, we have identified several directions to improve the performance of the error detection system.

Afterwards, we evaluated CEO with a small scale user-study. The explanations generated by CEO were described as relevant but difficult to understand. The number of explanations and their presentation were the problems as it required a high cognitive effort to read, study and understand the ten explanations. A point of confusion for the users was the consequence of deleting an assertion. It was not explicitly mentioned that the open-world assumption is employed and thus deleting an assertion does not imply that the assertion is false. In addition, a problem in the ranking of the counterfactuals was identified. The most abstract modifications were the best ranked but were never the preferred explanations. In Section 7.3.3, we attributed this issue to the choice of similarity metric that explores the class hierarchy depth-wise and thus favors abstract counterfactuals. A different similarity metric that favors classes at the same depth as the original class could address this issue. Finally, CEO does not yet handle the insertion of assertions and some assertion types (e.g. `DataPropertyAssertions` or `NegativeObjectPropertyAssertion`) which prevents the generation of some explanations that may be expected by the users.

These experiments showed the potential of OBIC and CEO as solutions for the design of an XIS. Despite issues regarding the error detection system, OBIC proved to be a promising framework to ensure that human-understandable concepts are used and that a formal logic is applied to control the behavior of the models. The counterfactual explanations generated by CEO are in line with the equivalent machine learning methods. Experts found the explanations useful and even enabled them to detect a design issue without any expertise in ontologies. We note several directions to further improve and evaluate these contributions. The experiments were conducted on a toy problem that does not correspond to a real application. This may hide particular issues that we are not aware of. The explanation interface of OBIC was not tested as the user-study focused on the counterfactual explanations from CEO. Therefore, a user-study on OBIC and its explanations should be conducted to assess our intuition that this system increases trust compared to a traditional machine learning system. Concerning CEO, the small sample size of the user study limits the generalization of our experiments This survey was a preliminary study to find issues in CEO before conducting a larger survey to get more significant data. In this larger survey, other methods may be evaluated to enable the comparison with CEO.

One of the main problems in XAI that appeared through these experiments is the need to adapt explanations to the task and audience of the AI system. We discussed that, in OBIC, the choice of thresholds and the design of the ontology are specific to the requirements of the task and the user. Likewise, for CEO, the choice of proximity metric including the cost of the elementary operations and the similarity between classes also depend on the task and the user's preferences. Our contributions allow for customization through the described choices which is an advantage as it allows the users to personalize their experience and increase their satisfaction and trust in the system. Nevertheless, this flexibility demands the users to make these choices which are complex and ask for a good understanding of the functioning of the proposed solutions. We have yet to create simple methodologies to help users determine the ideal choices for their needs in an automatic or semi-automatic manner.

Chapter 8

Conclusion and future work

In this chapter, we summarize and discuss the contributions presented in this thesis. Then, some perspectives to improve and build upon our work are presented.

Conclusion

The main goal of this thesis was to address a problem of XAI that is the design of explainability methods. To do so, we used symbolic approaches that have been identified as a promising direction to create explainable AI systems. Notably, ontologies are commonly regarded as ideal candidates for this purpose as they are able to represent notions used by human beings, are machine-readable and are built using description logics. We conducted a literature review of XAI and identified several open problems. The first issue that appeared is a lack of consensus regarding the vocabulary of XAI. Then, we observed that the neurosymbolic AI domain, that seeks to combine symbolic AI approaches with machine learning, is not exploring the explainability facet of these new AI systems. Consequently, we proposed the design of an explainable intelligent system as described by DARPA [15] that is centered around an ontology. Afterwards, we explored methods to explain this XIS and developed a technique to generate counterfactual explanations for ontologies. Finally, we evaluated the XIS and the counterfactual explanations on the task of classifying images of musical instruments.

Our first contribution discussed in Chapter 4 concerns the terminology of XAI. We identified important terms of XAI and defined them with regard to their use and definitions in the literature. The terminology is user-centered as explanations are specific to each user. We created an ontology that represents the concepts defined in this terminology and their relationships. This ontology can be employed to categorize AI systems. Likewise, we provided an ontology design pattern to define explanations in XAI, based on an ODP to define explanations in any field. Although the terminology is based on the definitions observed in the literature, it only reflects our understanding of the XAI vocabulary. Further discussions within the community should be conducted in order to reach a consensus, which may take several years. Still, this terminology removed ambiguities that may have occurred within our contributions. The user-centered nature of this terminology guided the design of the explainable intelligent system.

Following the terminology, we introduced the design of an XIS for image classification in Chapter 5 that exploits an ontology to build the model and explain the predictions. This XIS is designed to be explainable and transparent. Other attributes of an AI system described in the terminology depend on the model used for the classifiers. The explainable model, OBIC, builds models that are capable of detecting observable properties defined in the ontology. Observable properties are characteristics of concepts that can be detected in the data and are used in the definitions of these concepts. Then, an error detection system is applied by extracting the predictions of each model and testing their consistency with the ontology. This architecture enables the explanation of the outcome of the system by providing the detected properties that led to the final prediction. In addition, the error detection system provides a tool to help the user decide whether to trust the

prediction. The predictions and explanations are given via a prototype of explanation interface that displays all available information about the prediction and the error detection system. The unsupervised error detection system was evaluated in Chapter 7 and showed that using inconsistencies in the ontology to detect errors is a promising direction. The evaluation allowed us to better understand the behavior of OBIC and identify points for improvements. Notably, OBIC would benefit from a method to determine the model that is responsible of an inconsistency. Moreover, the performance of the error detection is impacted by the design of the ontology and the performance of each model. Regarding the explanations, we observed that they are not faithful to the functioning of the classifiers. Indeed, the actual prediction is done by one model that does not exploit the predictions of the property classifiers. Furthermore, the explanations given by the explanation interface are *raw explanations*, that only extracts information about the system without making the causes of the prediction clear and comprehensible to any user.

The last contribution directly addresses the lack of refined explanations for OBIC. It introduces the CEO method in Chapter 6 that brings counterfactual explanations for ontologies. Its main goal is to explain the outcome of a logical reasoner, such as the inference of new assertions or the detection of an inconsistency. It explores a graph of counterfactual explanations that are individuals of an ontology and identifies the explanations that are most similar to the original individual. It is designed to be applicable to most ontologies as a way to assist ontologists to debug an ontology as well as to explain the outcome of the logical reasoner to laypersons. The CEO method was also tested in Chapter 7, with the task of explaining the error detection system of OBIC. A user-study was conducted with domain experts to determine the quality and relevance of the explanations. The goals of the contributions were reached as it helped detect and fix design issues in the ontology while also successfully explaining the error detection system to the experts. Like OBIC, the user-study identified several points that need improvements, such as the proximity metric or the presentation of the explanations. We also remarked that in its current state, CEO does not scale with large ontologies.

The main goal of this thesis was to propose an explainable intelligent system that leverages the qualities of symbolic AI. We observed that there is no "one size fits all" technique that can adequately explain an AI system to any user. Consequently, we designed an XIS as an assembly of separate building blocks. Although the main architecture remains the same (following DARPA's schema), each building block can be replaced with another similar technique or improved by combining several techniques together. The needs of the task and target audience of the AI system dictate this choice of techniques. Research in XAI is about both designing these building blocks and creating methodologies to assemble these blocks. However, the current work is heavily focused on designing the building blocks. Our contributions are no exception to the rule as we have introduced two building blocks: OBIC as an explainable model with error detection and CEO as a method to explain ontologies. OBIC was designed to explain image classification tasks but we discussed in Section 5.4 that it may applicable to other types of data. CEO is intended to explain most ontologies although some assertions are not yet handled. Thus, OBIC and CEO are generic frameworks that can be used for a variety of tasks to explain a decision to different audiences. It emerged from our experiments that many choices are required to implement these techniques, choices that enable the customization of the explanations to the task and audience. Yet, we have not been able to determine a methodology to tailor OBIC and CEO for a specific task and audience.

AI systems that use machine learning models to make decisions have no prior knowledge of human concepts. When the model is being trained, it learns its own set of concepts that are appropriate to carry out the task. This learned set of concepts is rarely aligned with human concepts which renders the decision process impossible to understand for humans. Post hoc explainability methods focus on identifying a mapping between the model's concepts and human concepts but the outcome of these methods is usually unfaithful and cannot be applied in sensitive applications. Conversely, ontologies can be combined with machine learning models as a way to ensure that the models use human concepts extracted from the ontologies in the learning process. Hence, the concepts are already known and understood which facilitates the design of faithful explana-

tions. Moreover, ontologies use deductive reasoning to infer new facts based on prior knowledge. This type of reasoning can easily be explained by tracing back the premises of an inference. On the opposite, machine learning models apply a form of inductive reasoning which cannot be as easily explained since it does not follow a formal logical process. Humans use both inductive and deductive reasoning to make inferences. That is why combining ontologies and machine learning models is ideal to mimic human reasoning and as a result make the decision process more understandable to humans. OBIC was designed with the intention of replicating this combination of inductive and deductive reasoning. For instance, a human can explain that they saw a wooden chair by pointing out the presence of some concepts such as chair legs, a backrest and armrests as well as a texture of wood or a color specific to wood. All these concepts put together in a single object results in a wooden chair by applying a deductive reasoning based on prior knowledge. Yet, when a human is asked why they saw a particular color, they may not be able to provide a similar deduction. Instead, they will probably rely on inductive reasoning based on their experiences which cannot be properly explained to another person. Indeed, each human has a unique decision process to detect these basic concepts e.g. color, sound. OBIC uses inductive reasoning to detect basic concepts through machine learning models and then apply deductive reasoning based on human knowledge to infer a fact in accordance with the detected basic concepts. In summary, the combination of symbolic approaches with machine learning models have the potential to replicate human reasoning and to exploit human knowledge to make accurate and explainable decisions. This combination is already researched in the field of neurosymbolic AI. In this thesis, we proposed a neurosymbolic method dedicated to explainable AI. Our contributions fulfilled our goal of designing an explainable intelligent system that leverages symbolic AI. In the following section, we will discuss the perspectives to improve these contributions.

Future work

We have discussed in the previous section that we did not provide a methodology to build an XIS that is adapted to a specific task, audience and domain of application. To our knowledge, little research is done on this matter. The proposed terminology and ontology for AI systems can be applied to devise a methodology to choose models and XAI techniques in order to build an XIS. The terminology needs to be updated to take into account the latest developments in the field. It could also be expanded to include a terminology about the evaluation of explanations and XAI methods which may then be included in the methodology to build an XIS. A possible direction to begin the research of such a methodology is to choose a real task and collaborate with the targeted audience to build an XIS that meets their needs. Indeed, the main issue in our evaluation of OBIC and CEO is that the task did not correspond to any real needs and the development of the XIS was not done in cooperation with the targeted users. Applying OBIC and CEO to a task with clearly identified requirements would help making choices such as the thresholds for OBIC, the proximity metric in CEO or the implementation of additional explainability techniques to explain the outcome of OBIC.

The experiments on OBIC showed that the performance of the classifiers affect the error detection system. A method to improve the classifiers' performance may be to exploit the explanation interface to get humans to add labels to unknown data points and thus obtain more training data. In addition, a system that is able to identify the classifier that is the most likely to be the source of the inconsistency needs to be added. We discussed that the explanations provided by OBIC are unfaithful, because they imply that the property classifiers influence the classification, which is not the case. Therefore, the current architecture where the *global classifier* is the only classifier responsible for the main classification should be modified to render the explanations faithful to the system. A way to address this issue may be to remove the *global classifier* and find a deductive method to make the classification. It could be done by using the ontology to find a list of classes that are compatible with the properties detected. The classes would need to be sorted to give a final class. The CEO method could then be applied to explore what would happen to the final class

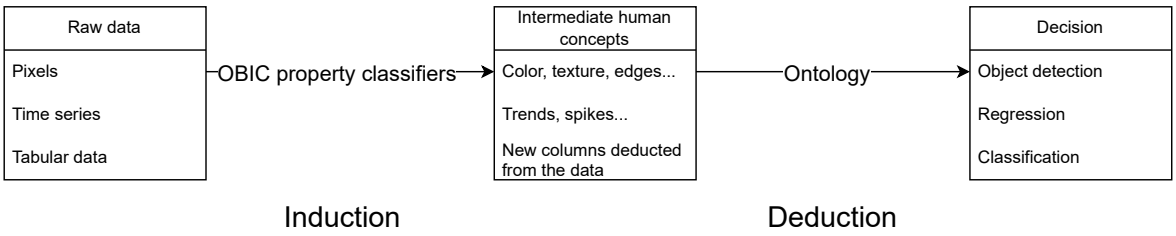


Figure 8.1: Diagram of a framework that generalizes OBIC to any type of task and data

if other properties were detected.

Still concerning OBIC, we discussed in Section 5.4 that it may be an instance of a more generic framework. We described this generic framework as following two steps: detect human concepts from raw data and apply logical reasoning on these concepts to make or confirm a decision. This process is illustrated in Figure 8.1. A transformation of raw data to intermediate human concepts is done with inductive algorithms (e.g. machine learning algorithms). These human concepts are determined by expert knowledge in the form of an ontology. Then, deductive reasoning is applied to make the final decision based on these concepts. OBIC is an instance of this framework where the raw data is pixels, the intermediate concepts are textures, shapes or more elaborate concepts such as the mechanisms of a musical instrument. Then, a logical reasoner is given these concepts in an ontology to verify a decision. Further investigations on this generic framework are to be conducted. Likewise, instantiations on other tasks and with different types of data will be explored.

Regarding CEO, we have already highlighted the most obvious perspectives in Section 6.5. The main perspectives are the expansion of the types of assertions handled, the addition of the assertion insertion operation in the heuristic search and the improvement of the algorithmic complexity. However, handling new assertions and adding a new of type of elementary operation has the direct effect of worsening the algorithmic complexity. A new method to explore counterfactuals is needed that allows the user to choose between longer execution times for more diversity or shorter execution time at the expense of potentially less relevant counterfactuals. On the subject of diversity, further work should be conducted to add this metric to the ranking of the counterfactuals in order to address some limitations that were observed in the evaluation of CEO in Chapter 7. Concerning the ranking of counterfactuals, the evaluation also highlighted the inadequacy of the proposed proximity metric. Specifically, the choice of similarity metric for classes of the ontology was the cause of this problem. Many similarity metrics have been proposed in the literature to measure the similarity of classes, that will be explored to address the problem on the proximity metric.

Finally, we note that the CEO method has opened up new opportunities to improve counterfactuals for machine learning. Our literature review exposed several issues in methods that generate counterfactuals for machine learning models. Mainly, the design of a proximity metric between categorical features and the identification of plausibility criteria. For the proximity metric in CEO, we have explored similarity metrics for individuals, assertions and classes of an ontology. We believe that these metrics could be applied to measure the similarity between categorical or ordinal features in the context of machine learning and thus solve the aforementioned problem. Similarly, the application of ontologies to determine plausibility criteria seems a promising lead that may be worth exploring.

Appendix A

CEO: Size of the search space

The search space Ω is defined as the set containing every possible counterfactuals of an individual. The current heuristic to explore this space does not insert new assertions to create counterfactuals. Therefore, the search space contains every combination of modification and deletion operations on the set of assertions of the original IKG. We note N the number of modifiable assertions of the original IKG i.e. every assertion except `ClassAssertions`. The total number of possible modifications for a single assertion is defined by the number of classes in the predicate's `range`. Let n_i be the number of classes in the predicate's `range` for the i-th assertion.

Let us consider the case where $N = 2$. The search space contains every possible modifications of the two assertions, every possible modification on one assertion and the deletion of the other and the deletion of both assertions.

$$|\Omega| = \underbrace{1}_{\text{Two deletions}} + \underbrace{n_1 + n_2}_{\text{Deletion of one assertion}} + \underbrace{n_1 \times n_2}_{\text{No deletion}} \tag{A.1}$$

Equation (A.1) shows the size of the search space when $N = 2$. There is only combination when every assertion is deleted, n_1 modifications when the second assertion is deleted, n_2 modifications when the first assertion is deleted and $n_1 \times n_2$ modifications when both assertions are kept. We can apply the same reasoning for an undefined number of assertions.

$$|\Omega| = \underbrace{1}_{\substack{\text{Deletion of all} \\ (N) \text{ assertions}}} + \underbrace{\sum_{i=1}^{N} n_i}_{\substack{\text{Deletion} \\ \text{of } N-1 \\ \text{assertions}}} + \underbrace{\sum_{(k_1,k_2)\in C_N^2} n_{k_1} \times n_{k_2}}_{\substack{\text{Deletion of} \\ N-2 \text{ assertions}}} + \cdots + \underbrace{\sum_{(k_1,\ldots,k_{N-1})\in C_N^{N-1}} \prod_{j=1}^{N-1} n_{k_j}}_{\substack{\text{Deletion of} \\ 1 \text{ assertion}}} + \underbrace{\prod_{j=1}^{N} n_j}_{\text{No deletion}} \tag{A.2}$$

Equation (A.2) shows the intuitive formula of the search space. The first term is always 1 and corresponds to the deletion of all assertions. The second term corresponds to every combination for the deletion of every assertion but one. The third term calculates every combination of two assertions for every possible pair of assertions, where C_N^k represents the set of combinations of k assertions picked from a set of N assertions. For instance, C_3^2 is the set of all possible unordered pairs picked from a set of three elements e.g. $\{(AB),(AC),(BC)\}$. In general, each term of Equation (A.2) corresponds to the number of possible counterfactuals for a given number of deleted assertions.

$$|\Omega| = \sum_{i=0}^{N} \sum_{(k_1,\ldots,k_i)\in C_N^i} \prod_{j=1}^{i} n_{k_j} \tag{A.3}$$

Equation (A.3) is the formula to calculate the size of the search space Ω. However, this formula is tricky to compute and we propose an alternative formula. We observe when rearranging Equation (A.2) that a pattern emerges as shown in Equation (A.4).

$$|\Omega| = 1 + n_1 + n_2\,(1 + n_1) + n_3\,(1 + n_1 + n_2\,(1 + n_1)) + \dots \tag{A.4}$$

Let σ be a series defined as:

$$\sigma\,(k) = \begin{cases} 1 & \text{if } k = 0 \\ 1 + n_1 & \text{if } k = 1 \\ n_k \sum_{i=1}^{k-1} \sigma\,(i) & \text{if } k > 1 \end{cases} \tag{A.5}$$

Theorem A.1.

$$\forall N \geq 1 \in \mathbb{N}, |\Omega| = \sum_{i=1}^{N} \sigma(i)$$

Proof. We will prove this statement by induction.

Base case For $N = 1$, $|\Omega| = 1 + n_1$.

$$\begin{aligned} \sum_{i=1}^{1} \sigma(i) &= \sigma(1) \\ &= 1 + n_1 \\ &= |\Omega| \end{aligned}$$

Inductive step Suppose the theorem holds for all values of N up to some t, $t \geq 1$. Let us verify the theorem for $N = t + 1$.

$$\begin{aligned} \sum_{i=1}^{t+1} \sigma(i) &= \sigma(t+1) + \sum_{i=1}^{t} \sigma(i) \\ &= n_{t+1} \sum_{i=1}^{t} \sigma(i) + \sum_{i=1}^{t} \sigma(i) \\ &= (1 + n_{t+1}) \sum_{i=1}^{t} \sigma(i) \\ &= (1 + n_{t+1}) \sum_{i=0}^{t} \sum_{(k_1,\dots,k_i) \in C_t^i} \prod_{j=1}^{i} n_{k_j} \\ &= \sum_{i=0}^{t} \left(\underbrace{\sum_{(k_1,\dots,k_i) \in C_t^i} \prod_{j=1}^{i} n_{k_j}}_{\text{Combinations without } (t+1)\text{-th assertion}} + \underbrace{n_{t+1} \sum_{(k_1,\dots,k_i) \in C_t^i} \prod_{j=1}^{i} n_{k_j}}_{\text{Combinations with at least the } (t+1)\text{-th assertion}} \right) \\ &= \sum_{i=0}^{t+1} \sum_{(k_1,\dots,k_i) \in C_{t+1}^i} \prod_{j=1}^{i} n_{k_j} \\ &= |\Omega| \end{aligned}$$

So the theorem holds for $N = t + 1$. By the principle of mathematical induction, the theorem holds for all $N \in \mathbb{N}$.

□

We can now compute the size of the search space by calculating and summing up each term of the series σ.

Appendix B

Definition of the musical instruments ontology

In this appendix, we describe the musical instruments ontology[1] used in Chapter 7. This ontology declares the same classes as the dataset given in this same chapter. Therefore, wooden chairs, wooden tables and brass pipes are also added to the ontology. Consequently, 4 atomic and disjoint classes are created that represent 4 different categories:

- The class `Instruments` defines the musical instrument, that is divided into several subclasses according to the Hornbostel-Sachs classification [223].
- The class `Furniture` contains the classes for wooden chairs and wooden tables.
- The class `Utilities` describes pipes and especially brass pipes.
- The class `Properties` defines several categories of properties that are used to define the classes of the dataset.

According to the requirements of OBIC (see Chapter 5), each class of the dataset must be defined with an `:observableProperty`. Hence, 5 subproperties of the object property `:observableProperty` are declared: `:hasApparentStrings`, `:hasMechanism`, `:hasMouthpiece`, `:hasShape` and `:hasTexture`. The range of each object property corresponds to a subclass of `Properties`. FFirst, we describe classes and object properties that are needed to define the utilities and furniture categories:

Shape The class `Shapes` is the range of `:hasShape`. Two distinct shapes are defined in the ontology: a bell shape and a violin shape. The domain of `:hasShape` is any class of the ontology since a shape can describe any element. These classes are defined as follows:

$$\texttt{Shape} \sqsubseteq \texttt{Properties} \tag{B.1}$$

$$\texttt{Bell} \sqsubseteq \texttt{Shape} \tag{B.2}$$

$$\texttt{ViolinShape} \sqsubseteq \texttt{Shape} \tag{B.3}$$

$$\tag{B.4}$$

Texture The class `Texture` is the range of `:hasTexture`. Two textures are defined that correspond to common textures of musical instruments: brass metal and wood. Similar to shapes, the domain of `:hasTexture` is any class of the ontology. Indeed, the definitions of wooden

[1] This ontology is available at `https://git.litislab.fr/s4xai/ontology-based-image-classifier/-/blob/main/data/instruments.owl`

chairs, wooden tables and brass pipes solely rely on this property.

$$\texttt{Texture} \sqsubseteq \texttt{Properties} \tag{B.5}$$
$$\texttt{BrassMetal} \sqsubseteq \texttt{Texture} \tag{B.6}$$
$$\texttt{Wood} \sqsubseteq \texttt{Texture} \tag{B.7}$$
$$\tag{B.8}$$

With these class and object properties descriptions, we can define the furniture and utilities classes:

$$\texttt{WoodenChair} \sqsubseteq \texttt{Furniture} \sqcap \exists \texttt{hasTexture.Wood} \sqcap \neg(\exists \texttt{hasTexture.BrassMetal}) \tag{B.9}$$
$$\texttt{WoodenTable} \sqsubseteq \texttt{Furniture} \sqcap \exists \texttt{hasTexture.Wood} \sqcap \neg(\exists \texttt{hasTexture.BrassMetal}) \tag{B.10}$$
$$\texttt{Pipes} \sqsubseteq \texttt{Utilities} \tag{B.11}$$
$$\texttt{BrassPipes} \sqsubseteq \texttt{Pipes} \sqcap \exists \texttt{hasTexture.BrassMetal} \sqcap \neg(\exists \texttt{hasTexture.Wood}) \tag{B.12}$$

Furthermore, `WoodenChair` is disjoint with `WoodenTable`, `Utilities`, `Properties`, `Furniture` and `Instruments` are disjoint.

We now focus on describing the musical instruments. According to the Hornbostel-Sachs classification [223], instruments can be subdivided into several families. For our application, we retained 17 instruments: cello, contrabass, viola, violin, guitar, harp, harpsichord, piano, french horn, serpent, trombone, trumpet, tuba, bassoon, clarinet, oboe and saxophone. These instruments belong to either the strings family or the winds family. The string family is divided into three sub-families that are bowed, plucked or struck strings. The wind family is divided into two sub-families, brass and woodwinds.

Every musical instrument has one or several mechanisms that that are visible and allow them to make a sound, tune the instrument etc. For instance, the keyboard and the pedals of a piano are considered mechanisms since they are usually visible and are used when playing the instrument. Some mechanisms are unique to families of instruments; considering the chosen instruments, we identified mechanisms that are specific to strings and other specific to winds. Two exceptions are a keyboard and pedals that are mostly found on string instruments but some wind instruments also have them (e.g. organ, accordion, melodica). String instruments have two specific mechanisms: strings and pegs. Wind instruments have 3 mechanisms: keys, pistons and slide. However, pistons and slide are specific to brass instruments. Based on these observations, we define the class `Mechanism` as the range of the observable property `:hasMechanism`, `Instruments` are the domain of this property. The subclasses of `Mechanism` are defined as follows:

$$\texttt{Mechanism} \sqsubseteq \texttt{Properties} \tag{B.13}$$
$$\texttt{Keyboard} \sqsubseteq \texttt{Mechanism} \tag{B.14}$$
$$\texttt{Pedals} \sqsubseteq \texttt{Mechanism} \tag{B.15}$$
$$\texttt{StringsMechanism} \sqsubseteq \texttt{Mechanism} \tag{B.16}$$
$$\texttt{WindsMechanism} \sqsubseteq \texttt{Mechanism} \tag{B.17}$$
$$\texttt{Pegs} \sqsubseteq \texttt{StringsMechanism} \tag{B.18}$$
$$\texttt{String} \sqsubseteq \texttt{StringsMechanism} \tag{B.19}$$
$$\texttt{Keys} \sqsubseteq \texttt{WindsMechanism} \tag{B.20}$$
$$\texttt{BrassMechanism} \sqsubseteq \texttt{WindsMechanism} \tag{B.21}$$
$$\texttt{Pistons} \sqsubseteq \texttt{BrassMechanism} \tag{B.22}$$
$$\texttt{Slide} \sqsubseteq \texttt{BrassMechanism} \tag{B.23}$$

The object property `:hasApparentStrings` is uses to describe instruments with strings that are not hidden e.g. violin, viola, harp. Consequently, its range is the mechanism `String` and its domain is limited to `Strings`.

Finally, one particular feature of the chosen wind instruments is that they necessarily have a mouthpiece, in the form of a reed for woodwinds and a brass mouthpiece for brass instruments. A mouthpiece is usually visible on a musical instrument and leads to the final set of properties. Moreover, reeds are made of wood which can be added to the description of a reed.

$$\texttt{Mouthpiece} \sqsubseteq \texttt{Properties} \tag{B.24}$$

$$\texttt{BrassMouthpiece} \sqsubseteq \texttt{Mouthpiece} \tag{B.25}$$

$$\texttt{Reed} \sqsubseteq \texttt{Mouthpiece} \sqcap \exists \texttt{hasTexture.Wood} \tag{B.26}$$

The range of the object property :hasMouthpiece is Mouthpiece and its domain is Winds. This property is functional since the described wind instruments only have one mouthpiece.

With the declaration of all the properties, we can describe the classes that represent each chosen instrument. Every class that represents an instrument is disjoint with the other instruments. First, we describe the string instruments family.

$$\texttt{Strings} \sqsubseteq \texttt{Instruments} \sqcap \neg (\exists \texttt{hasMechanism.WindsMechanism}) \sqcap \neg \left(\exists \texttt{hasShape.Bell}\right) \tag{B.27}$$

$$\texttt{Bowed} \sqsubseteq \texttt{Strings} \sqcap \neg \left(\exists \texttt{hasMechanism.Keyboard}\right) \tag{B.28}$$

$$\texttt{Plucked} \sqsubseteq \texttt{Strings} \tag{B.29}$$

$$\texttt{Struck} \sqsubseteq \texttt{Strings} \tag{B.30}$$

$$\texttt{ViolinFamily} \equiv \texttt{Bowed} \sqcap \exists \texttt{hasApparentStrings.String} \sqcap \exists \texttt{hasMechanism.Pegs} \sqcap \exists \texttt{hasMechanism.String} \sqcap \exists \texttt{hasShape.ViolinShape} \sqcap \exists \texttt{hasTexture.Wood} \tag{B.31}$$

$$\texttt{Cello} \sqsubseteq \texttt{ViolinFamily} \tag{B.32}$$

$$\texttt{Contrabass} \sqsubseteq \texttt{ViolinFamily} \tag{B.33}$$

$$\texttt{Violin} \sqsubseteq \texttt{ViolinFamily} \tag{B.34}$$

$$\texttt{Viola} \sqsubseteq \texttt{ViolinFamily} \tag{B.35}$$

$$\texttt{Guitar} \sqsubseteq \texttt{Plucked} \sqcap \exists \texttt{hasApparentStrings.String} \sqcap \exists \texttt{hasMechanism.Pegs} \sqcap \exists \texttt{hasMechanism.String} \sqcap \exists \texttt{hasTexture.Wood} \sqcap \neg \left(\exists \texttt{hasMechanism.Keyboard}\right) \tag{B.36}$$

$$\texttt{Harp} \sqsubseteq \texttt{Plucked} \sqcap \exists \texttt{hasApparentStrings.String} \sqcap \exists \texttt{hasMechanism.Pedals} \sqcap \exists \texttt{hasMechanism.String} \sqcap \neg \left(\exists \texttt{hasMechanism.Keyboard}\right) \tag{B.37}$$

$$\texttt{Harpsichord} \sqsubseteq \texttt{Plucked} \sqcap \exists \texttt{hasMechanism.Keyboard} \sqcap \exists \texttt{hasTexture.Wood} \sqcap \neg (\exists \texttt{hasMechanism.Pedals}) \tag{B.38}$$

$$\texttt{Piano} \sqsubseteq \texttt{Struck} \sqcap \exists \texttt{hasMechanism.Keyboard} \sqcap \exists \texttt{hasMechanism.Pedals} \sqcap \exists \texttt{hasTexture.Wood} \tag{B.39}$$

Afterwards, we define the wind instruments family. We note for the definition of Winds that since :hasMouthpiece is functional, the existential quantification that implies that it has some

mouthpiece is equivalent to the number restriction imposing that it has exactly one mouthpiece.

$$\texttt{Winds} \equiv \texttt{Instruments} \sqcap \exists \texttt{hasMouthpiece}.\texttt{Mouthpiece} \tag{B.40}$$

$$\texttt{Winds} \sqsubseteq \neg(\exists \texttt{hasMechanism}.\texttt{Pedals}) \sqcap \neg\left(\exists \texttt{hasMechanism}.\texttt{StringsMechanism}\right) \tag{B.41}$$

$$\texttt{Brass} \equiv \texttt{Winds} \sqcap \forall \texttt{hasMouthpiece}.\texttt{BrassMouthpiece} \tag{B.42}$$

$$\texttt{Brass} \sqsubseteq \neg\left(\exists \texttt{hasMechanism}.\texttt{Keyboard}\right) \tag{B.43}$$

$$\begin{aligned}\texttt{Woodwinds} \sqsubseteq\ & \texttt{Winds} \sqcap \neg(\exists \texttt{hasMechanism}.\texttt{BrassMechanism}) \\ & \sqcap \neg\left(\exists \texttt{hasMechanism}.\texttt{Keyboard}\right) \sqcap \neg\left(\exists \texttt{hasMouthpiece}.\texttt{BrassMouthpiece}\right)\end{aligned} \tag{B.44}$$

$$\begin{aligned}\texttt{FrenchHorn} \sqsubseteq\ & \texttt{Brass} \sqcap \exists \texttt{hasMechanism}.\texttt{Pistons} \sqcap \exists \texttt{hasShape}.\texttt{Bell} \\ & \sqcap \exists \texttt{hasTexture}.\texttt{BrassMetal} \sqcap \neg(\exists \texttt{hasMechanism}.\texttt{Slide}) \\ & \sqcap \neg(\exists \texttt{hasTexture}.\texttt{Wood})\end{aligned} \tag{B.45}$$

$$\begin{aligned}\texttt{Serpent} \sqsubseteq\ & \texttt{Brass} \sqcap \exists \texttt{hasMechanism}.\texttt{Keys} \sqcap \exists \texttt{hasTexture}.\texttt{Wood} \\ & \sqcap \neg(\exists \texttt{hasMechanism}.\texttt{Pistons}) \sqcap \neg(\exists \texttt{hasTexture}.\texttt{BrassMetal}) \\ & \sqcap \neg(\exists \texttt{hasMechanism}.\texttt{Slide})\end{aligned} \tag{B.46}$$

$$\begin{aligned}\texttt{Trombone} \sqsubseteq\ & \texttt{Brass} \sqcap \exists \texttt{hasMechanism}.\texttt{Slide} \sqcap \exists \texttt{hasShape}.\texttt{Bell} \\ & \sqcap \exists \texttt{hasTexture}.\texttt{BrassMetal} \sqcap \neg\left(\exists \texttt{hasMechanism}.\texttt{Keys}\right) \\ & \sqcap \neg(\exists \texttt{hasMechanism}.\texttt{Pistons}) \sqcap \neg(\exists \texttt{hasTexture}.\texttt{Wood})\end{aligned} \tag{B.47}$$

$$\begin{aligned}\texttt{Trumpet} \sqsubseteq\ & \texttt{Brass} \sqcap \exists \texttt{hasMechanism}.\texttt{Pistons} \sqcap \exists \texttt{hasShape}.\texttt{Bell} \\ & \sqcap \exists \texttt{hasTexture}.\texttt{BrassMetal} \sqcap \neg\left(\exists \texttt{hasMechanism}.\texttt{Keys}\right) \\ & \sqcap \neg(\exists \texttt{hasTexture}.\texttt{Wood})\end{aligned} \tag{B.48}$$

$$\begin{aligned}\texttt{Tuba} \sqsubseteq\ & \texttt{Brass} \sqcap \exists \texttt{hasMechanism}.\texttt{Pistons} \sqcap \exists \texttt{hasShape}.\texttt{Bell} \\ & \sqcap \exists \texttt{hasTexture}.\texttt{BrassMetal} \sqcap \neg\left(\exists \texttt{hasMechanism}.\texttt{Keys}\right) \\ & \sqcap \neg(\exists \texttt{hasMechanism}.\texttt{Slide}) \sqcap \neg(\exists \texttt{hasTexture}.\texttt{Wood})\end{aligned} \tag{B.49}$$

$$\begin{aligned}\texttt{Bassoon} \sqsubseteq\ & \texttt{Woodwinds} \sqcap \exists \texttt{hasMechanism}.\texttt{Keys} \sqcap \exists \texttt{hasMouthpiece}.\texttt{Reed} \\ & \sqcap \exists \texttt{hasTexture}.\texttt{Wood} \sqcap \neg(\exists \texttt{hasTexture}.\texttt{BrassMetal})\end{aligned} \tag{B.50}$$

$$\begin{aligned}\texttt{Clarinet} \sqsubseteq\ & \texttt{Woodwinds} \sqcap \exists \texttt{hasMechanism}.\texttt{Keys} \sqcap \exists \texttt{hasShape}.\texttt{Bell} \\ & \sqcap \exists \texttt{hasMouthpiece}.\texttt{Reed} \sqcap \exists \texttt{hasTexture}.\texttt{Wood} \\ & \sqcap \neg(\exists \texttt{hasTexture}.\texttt{BrassMetal})\end{aligned} \tag{B.51}$$

$$\begin{aligned}\texttt{Oboe} \sqsubseteq\ & \texttt{Woodwinds} \sqcap \exists \texttt{hasMechanism}.\texttt{Keys} \sqcap \exists \texttt{hasShape}.\texttt{Bell} \\ & \sqcap \exists \texttt{hasMouthpiece}.\texttt{Reed} \sqcap \exists \texttt{hasTexture}.\texttt{Wood} \\ & \sqcap \neg(\exists \texttt{hasTexture}.\texttt{BrassMetal})\end{aligned} \tag{B.52}$$

$$\begin{aligned}\texttt{Saxophone} \sqsubseteq\ & \texttt{Woodwinds} \sqcap \exists \texttt{hasMechanism}.\texttt{Keys} \sqcap \exists \texttt{hasShape}.\texttt{Bell} \\ & \sqcap \exists \texttt{hasMouthpiece}.\texttt{Reed} \sqcap \exists \texttt{hasTexture}.\texttt{BrassMetal}\end{aligned} \tag{B.53}$$

Bibliography

[1] Henry A. Kautz. "The third AI summer: AAAI Robert S. Engelmore Memorial Lecture". In: *AI Magazine* 43.1 (2022), pp. 105–125. DOI: 10.1002/aaai.12036 (cit. on pp. 1, 2, 23, 24).

[2] Pramila P. Shinde and Seema Shah. "A Review of Machine Learning and Deep Learning Applications". In: *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*. IEEE, 2018. DOI: 10.1109/iccubea.2018.8697857 (cit. on pp. 1, 23).

[3] Tom Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf (cit. on pp. 1, 23).

[4] Cornelius T. Leondes. *Expert Systems. The Technology of Knowledge Management for the 21st Century Six Volume Set*. Academic Press, 2001, p. 2200. ISBN: 9780124438804 (cit. on pp. 2, 23).

[5] K. P. Tripathi. "A review on knowledge-based expert system: concept and architecture". In: *IJCA Special Issue on Artificial Intelligence Techniques-Novel Approaches & Practical Applications* 4 (2011), pp. 19–23 (cit. on pp. 2, 23).

[6] John P McDermott. "RI: an Expert in the Computer Systems Domain." In: *AAAI*. Vol. 1. 1980, pp. 269–271 (cit. on pp. 2, 24).

[7] David C. Brock. "Learning from Artificial Intelligence's Previous Awakenings: The History of Expert Systems". In: *AI Magazine* 39.3 (2018), pp. 3–15. DOI: 10.1609/aimag.v39i3.2809 (cit. on pp. 2, 24).

[8] Julia Angwin et al. "Machine bias. There's software used across the country to predict future criminals. And it's biased against blacks." In: *ProPublica* (2016). URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (visited on 17/02/2023) (cit. on pp. 2, 24).

[9] European Commission. *General Data Protection Regulation*. European Union, 2016. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679 (cit. on pp. 2, 24).

[10] Artur d'Avila Garcez and Luis C. Lamb. "Neurosymbolic AI: the 3rd wave". In: *Artificial Intelligence Review* (2023). DOI: 10.1007/s10462-023-10448-w (cit. on pp. 2, 24).

[11] Tim Miller, Piers Howe and Liz Sonenberg. "Explainable AI: Beware of Inmates Running the Asylum Or: How I Learnt to Stop Worrying and Love the Social and Behavioural Sciences". In: *IJCAI-17 Workshop on Explainable AI (XAI) Proceedings*. 2017. DOI: 10.48550/ARXIV.1712.00547. eprint: 1712.00547 (cs.AI) (cit. on pp. 3, 25, 40, 41, 83).

[12] Markus Langer et al.
"What do we want from Explainable Artificial Intelligence (XAI)? – A stakeholder perspective on XAI and a conceptual model guiding interdisciplinary XAI research".
In: *Artificial Intelligence* 296 (2021), p. 103473. DOI: 10.1016/j.artint.2021.103473 (cit. on pp. 3, 25).

[13] Jan Maarten Schraagen et al. "Trusting the X in XAI: Effects of different types of explanations by a self-driving car on trust, explanation satisfaction and mental models".
In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 64.1 (2020), pp. 339–343. DOI: 10.1177/1071181320641077 (cit. on pp. 3, 25, 39, 46, 47).

[14] Arun Das and Paul Rad.
"Opportunities and Challenges in Explainable Artificial Intelligence (XAI): A Survey".
In: (2020). DOI: 10.48550/ARXIV.2006.11371. arXiv: 2006.11371 [cs.CV] (cit. on pp. 3, 25).

[15] David Gunning. "Explainable artificial intelligence (xai)".
In: *Defense Advanced Research Projects Agency (DARPA), nd Web* 2.2 (2017) (cit. on pp. 3, 6, 7, 15, 18, 25, 27, 28, 41, 48, 60, 75, 76, 83, 92, 96, 147).

[16] David Gunning and David Aha.
"DARPA's Explainable Artificial Intelligence (XAI) Program".
In: *AI Magazine* 40.2 (2019), pp. 44–58. DOI: 10.1609/aimag.v40i2.2850 (cit. on pp. 3, 25).

[17] Ross Taylor et al. "Galactica: A Large Language Model for Science". In: (2022). DOI: 10.48550/ARXIV.2211.09085. arXiv: 2211.09085 [cs.CL] (cit. on pp. 3, 25).

[18] Will Douglas Heaven. "Why Meta's latest large language model survived only three days online. Galactica was supposed to help scientists. Instead, it mindlessly spat out biased and incorrect nonsense." In: *MIT Technology Review*. Artificial Intelligence (2022). URL: https://www.technologyreview.com/2022/11/18/1063487/meta-large-language-model-ai-only-survived-three-days-gpt-3-science (visited on 14/03/2023) (cit. on pp. 3, 25).

[19] Ryan Browne. "All you need to know about ChatGPT, the A.I. chatbot that's got the world talking and tech giants clashing ". In: *CNBC* (2023). URL: https://www.cnbc.com/2023/02/08/what-is-chatgpt-viral-ai-chatbot-at-heart-of-microsoft-google-fight.html (visited on 01/03/2023) (cit. on pp. 3, 25).

[20] Chris Stokel-Walker.
"ChatGPT listed as author on research papers: many scientists disapprove".
In: *Nature* (2023). URL: https://www.nature.com/articles/d41586-023-00107-z (visited on 25/07/2023) (cit. on pp. 3, 25).

[21] OpenAI. *Introducing ChatGPT*. Ed. by OpenAI. OpenAI. 2022. URL: https://openai.com/blog/chatgpt (visited on 01/03/2023) (cit. on pp. 4, 25).

[22] Billy Perrigo.
"OpenAI Used Kenyan Workers on Less Than $2 Per Hour to Make ChatGPT Less Toxic".
In: *TIME* (2023).
URL: https://time.com/6247678/openai-chatgpt-kenya-workers/ (visited on 27/02/2023) (cit. on pp. 4, 25).

[23] Independent High-Level Expert Group On AI. *Ethics Guidelines for Trustworthy AI*. European Commission, 2019.
URL: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai (visited on 02/03/2023) (cit. on pp. 4, 25).

[24] Google. *AI at Google: our principles*. Google. 2018. URL: https://www.blog.google/technology/ai/ai-principles/ (visited on 02/03/2023) (cit. on pp. 4, 25).

[25] Cade Metz. "Seeking Ground Rules for AI". In: *New York Times* (2019). URL: https://www.nytimes.com/2019/03/01/business/ethical-ai-recommendations.html (visited on 02/03/2023) (cit. on pp. 4, 25).

[26] Jessica Fjeld et al. "Principled Artificial Intelligence: Mapping Consensus in Ethical and Rights-Based Approaches to Principles for AI". In: *SSRN Electronic Journal* (2020). DOI: 10.2139/ssrn.3518482 (cit. on pp. 4, 25, 26).

[27] Patrick Mikalef et al. "Thinking responsibly about responsible AI and 'the dark side' of AI". In: *European Journal of Information Systems* 31.3 (2022), pp. 257–268. DOI: 10.1080/0960085x.2022.2026621 (cit. on pp. 4, 25, 66).

[28] 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. DOI: 10.1016/j.inffus.2019.12.012 (cit. on pp. 4, 11, 14, 26, 32, 35, 38, 57–62, 68).

[29] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8466590 (cit. on pp. 5, 11, 14, 26, 35, 38, 57, 58, 61–63, 67, 75).

[30] David Gunning et al. "DARPA's explainable AI (XAI) program: A retrospective". In: *Applied AI Letters* 2.4 (2021). DOI: 10.1002/ail2.61 (cit. on pp. 5, 26).

[31] Freddy Lecue. "On the role of knowledge graphs in explainable AI". In: *Semantic Web* 11.1 (2020). Ed. by Pascal Hitzler and Krzysztof Janowicz, pp. 41–51. DOI: 10.3233/sw-190374. URL: https://dl.acm.org/doi/abs/10.3233/SW-190374 (cit. on pp. 7, 28, 38, 82, 106, 109).

[32] Federico Cabitza et al. "Quod erat demonstrandum? - Towards a typology of the concept of explanation for the design of explainable AI". In: *Expert Systems with Applications* 213 (2023), p. 118888. DOI: 10.1016/j.eswa.2022.118888 (cit. on pp. 8, 32, 33, 62).

[33] Tim Miller. "Explanation in artificial intelligence: Insights from the social sciences". In: *Artificial Intelligence* 267 (2019), pp. 1–38. DOI: 10.1016/j.artint.2018.07.007 (cit. on pp. 8, 9, 32–34, 39, 59, 99, 101).

[34] Vijay Arya et al. "One Explanation Does Not Fit All: A Toolkit And Taxonomy Of AI Explainability Techniques". In: *INFORMS Annual Meeting*. 2021. URL: https://arxiv.org/abs/1909.03012 (cit. on pp. 8, 11, 32, 35, 41, 58, 60–62, 76, 83).

[35] Shruthi Chari et al. "Explanation Ontology: A Model of Explanations for User-Centered AI". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 228–243. DOI: 10.1007/978-3-030-62466-8_15 (cit. on pp. 8, 32).

[36] Juan M. Durán. "Dissecting scientific explanation in AI (sXAI): A case for medicine and healthcare". In: *Artificial Intelligence* 297 (2021), p. 103498. DOI: 10.1016/j.artint.2021.103498 (cit. on pp. 9, 33, 34).

[37] Robert R. Hoffman and Gary Klein.
"Explaining Explanation, Part 1: Theoretical Foundations".
In: *IEEE Intelligent Systems* 32.3 (2017), pp. 68–73. DOI: 10.1109/mis.2017.54
(cit. on pp. 9, 32–34, 61).

[38] Atocha Aliseda. *Abductive Reasoning Logical Investigations into Discovery and Explanation. Logical Investigations into Discovery and Explanation.*
Springer Dordrecht, 2006. ISBN: 9781402039072. DOI: 10.1007/1-4020-3907-7
(cit. on pp. 9, 34).

[39] Sandra Wachter, Brent Mittelstadt and Chris Russell. "Counterfactual explanations without opening the black box: Automated decisions and the GDPR".
In: *Harv. JL & Tech.* 31 (2017), p. 841.
URL: https://arxiv.org/ftp/arxiv/papers/1711/1711.00399.pdf
(cit. on pp. 9, 12, 34, 99–104).

[40] Fan Yang, Mengnan Du and Xia Hu.
"Evaluating Explanation Without Ground Truth in Interpretable Machine Learning".
In: (2019). DOI: 10.48550/ARXIV.1907.06831. arXiv: 1907.06831 [cs.LG]
(cit. on pp. 10, 39–43, 62, 68).

[41] Jianlong Zhou et al. "Evaluating the Quality of Machine Learning Explanations: A Survey on Methods and Metrics". In: *Electronics* 10.5 (2021), p. 593.
DOI: 10.3390/electronics10050593 (cit. on pp. 10, 39–43, 62).

[42] Meike Nauta et al. "From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI". In: *ACM Computing Surveys* (2023).
DOI: 10.1145/3583558 (cit. on pp. 10, 39–44, 62, 63, 69).

[43] Maximilian Förster et al.
"Evaluating explainable Artifical intelligence - What users really appreciate".
In: *In Proceedings of the 28th European Conference on Information Systems (ECIS)*. 2020.
URL: https://web.archive.org/web/20220803134652id_/https://aisel.aisnet.org/cgi/viewcontent.cgi?article=1194&context=ecis2020_rp
(cit. on pp. 10, 39–42, 45, 46, 62–64).

[44] Timo Speith.
"A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods".
In: *2022 ACM Conference on Fairness, Accountability, and Transparency*. ACM, 2022.
DOI: 10.1145/3531146.3534639 (cit. on pp. 10, 11, 35, 36, 62).

[45] Riccardo Guidotti et al. "A Survey of Methods for Explaining Black Box Models".
In: *ACM Computing Surveys* 51.5 (2018), pp. 1–42. DOI: 10.1145/3236009
(cit. on pp. 11, 12, 32, 35, 59–62, 68, 102).

[46] Nadia Burkart and Marco F. Huber.
"A Survey on the Explainability of Supervised Machine Learning".
In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 245–317.
DOI: 10.1613/jair.1.12228 (cit. on pp. 11, 32, 35, 38, 62).

[47] Christian Schorr et al. "Neuroscope: An Explainable AI Toolbox for Semantic Segmentation and Image Classification of Convolutional Neural Nets".
In: *Applied Sciences* 11.5 (2021), p. 2199. DOI: 10.3390/app11052199
(cit. on pp. 11, 78, 84).

[48] Vitali Petsiuk et al. "Black-Box Explanation of Object Detectors via Saliency Maps".
In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 11443–11452 (cit. on pp. 11, 78).

[49] Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin. ""Why Should I Trust You?"" In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016. DOI: 10.1145/2939672.2939778 (cit. on pp. 11, 35, 37, 44, 59–61, 78).

[50] Scott M. Lundberg and Su-In Lee. "A Unified Approach to Interpreting Model Predictions". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf (cit. on pp. 11, 35, 78).

[51] Avanti Shrikumar, Peyton Greenside and Anshul Kundaje. "Learning Important Features Through Propagating Activation Differences". In: *ICML'17: Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 3145–3153. URL: https://proceedings.mlr.press/v70/shrikumar17a.html (cit. on pp. 11, 35, 78).

[52] Julius Adebayo et al. "Sanity Checks for Saliency Maps". In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31. Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/294a8ed24b1ad22ec2e7efea049b8737-Paper.pdf (cit. on pp. 11, 12, 79).

[53] Karen Simonyan, Andrea Vedaldi and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2014. URL: http://arxiv.org/abs/1312.6034 (cit. on pp. 11, 79).

[54] Christoph Molnar. *Interpretable Machine Learning A Guide for Making Black Box Models Explainable. A Guide for Making Black Box Models Explainable*. 2022. URL: https://christophm.github.io/interpretable-ml-book/ (cit. on pp. 11, 79, 100).

[55] Emmanuel Pintelas et al. "Explainable Machine Learning Framework for Image Classification Problems: Case Study on Glioma Cancer Prediction". In: *Journal of Imaging* 6.6 (2020), p. 37. DOI: 10.3390/jimaging6060037 (cit. on pp. 12, 80–83, 86, 96).

[56] Yuxia Geng et al. "Human-centric Transfer Learning Explanation via Knowledge Graph". In: *AAAI-19 Workshop on Network Interpretability for Deep Learning*. Vol. 27. 2019, p. 28 (cit. on pp. 12, 82).

[57] Tanguy Pommellet and Freddy Lecue. "Feeding machine learning with knowledge graphs for explainable object detection". In: *CEUR Workshop Proceedings*. Vol. 2456. 2019, pp. 277–280. URL: https://web.archive.org/web/20210616123034id_/http://ceur-ws.org/Vol-2456/paper72.pdf (cit. on pp. 12, 82).

[58] Kenneth Marino, Ruslan Salakhutdinov and Abhinav Gupta. "The More You Know: Using Knowledge Graphs for Image Classification". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017. DOI: 10.1109/cvpr.2017.10 (cit. on pp. 12, 82).

[59] Ramaravind K. Mothilal, Amit Sharma and Chenhao Tan.
"Explaining machine learning classifiers through diverse counterfactual explanations".
In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*.
ACM, 2020. DOI: 10.1145/3351095.3372850 (cit. on pp. 12, 13, 102–106).

[60] Sahil Verma, John Dickerson and Keegan Hines.
"Counterfactual Explanations for Machine Learning: A Review".
In: *ML-RSA @ NeurIPS2020* (2020). arXiv: 2010.10596 [cs.LG] (cit. on pp. 12, 100–104).

[61] Ilia Stepin et al. "A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence". In: *IEEE Access* 9 (2021), pp. 11974–12001.
DOI: 10.1109/access.2021.3051315 (cit. on pp. 13, 34, 100–103, 105).

[62] Mark T. Keane et al. "If Only We Had Better Counterfactual Explanations: Five Key Deficits to Rectify in the Evaluation of Counterfactual XAI Techniques". In: *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI-21), August, 2021* (2021).
arXiv: 2103.01035 [cs.LG] (cit. on pp. 13, 101, 102, 105).

[63] Zachary C. Lipton. "The Mythos of Model Interpretability".
In: *Queue* 16.3 (2018), pp. 31–57. DOI: 10.1145/3236386.3241340
(cit. on pp. 14, 57, 59, 60, 62, 65).

[64] *DOLCE+DnS Ultralite Ontology*.
URL: http://www.ontologydesignpatterns.org/ont/dul/DUL.owl (visited on 19/09/2023) (cit. on pp. 14, 64, 69, 72, 74).

[65] Ilaria Tiddi, Mathieu d'Aquin and Enrico Motta.
"An Ontology Design Pattern to Define Explanations".
In: *Proceedings of the 8th International Conference on Knowledge Capture*. ACM, 2015.
DOI: 10.1145/2815833.2815844 (cit. on pp. 14, 32, 58, 64, 71–74).

[66] Stathis Psillos. "Past and Contemporary Perspectives on Explanation".
In: *General Philosophy of Science*. Elsevier, 2007, pp. 97–173.
DOI: 10.1016/b978-044451548-3/50004-5 (cit. on p. 32).

[67] Finale Doshi-Velez and Been Kim.
"Towards A Rigorous Science of Interpretable Machine Learning". In: (2017).
DOI: 10.48550/ARXIV.1702.08608. arXiv: 1702.08608 [stat.ML]
(cit. on pp. 32, 43, 58, 61, 62, 68).

[68] Robert R. Hoffman, Shane T. Mueller and Gary Klein.
"Explaining Explanation, Part 2: Empirical Foundations".
In: *IEEE Intelligent Systems* 32.4 (2017), pp. 78–86. DOI: 10.1109/mis.2017.3121544
(cit. on pp. 32, 61).

[69] Gesina Schwalbe and Bettina Finzel. "A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts".
In: *Data Mining and Knowledge Discovery* (2023). DOI: 10.1007/s10618-022-00867-8
(cit. on pp. 35, 36, 44, 62).

[70] Sebastian Bach et al. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation".
In: *PLOS ONE* 10.7 (2015). Ed. by Oscar Deniz Suarez, e0130140.
DOI: 10.1371/journal.pone.0130140 (cit. on pp. 35, 44).

[71] 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.
DOI: 10.1038/s42256-019-0048-x (cit. on p. 37).

[72] Sina Mohseni, Niloofar Zarei and Eric D. Ragan. “A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems”.
In: *ACM Transactions on Interactive Intelligent Systems* 11.3-4 (2021), pp. 1–45.
DOI: 10.1145/3387166 (cit. on p. 37).

[73] Anna Hedström et al. “Quantus: An Explainable AI Toolkit for Responsible Evaluation of Neural Network Explanations and Beyond”.
In: *Journal of Machine Learning Research* 24.34 (2023), pp. 1–11.
URL: http://jmlr.org/papers/v24/22-0142.html (cit. on pp. 37, 76).

[74] Umang Bhatt, Adrian Weller and José M. F. Moura.
“Evaluating and Aggregating Feature-based Model Explanations”. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*.
International Joint Conferences on Artificial Intelligence Organization, 2020.
DOI: 10.24963/ijcai.2020/417 (cit. on pp. 37, 39, 44, 45, 63).

[75] David Alvarez Melis and Tommi Jaakkola.
“Towards Robust Interpretability with Self-Explaining Neural Networks”.
In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31.
Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper/2018/file/3e9f0fc9b2f89e043bc6233994dfcf76-Paper.pdf (cit. on pp. 37, 44, 60, 63, 78).

[76] Been Kim et al. “Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)”.
In: *ICML’18: Proceedings of the 35th International Conference on Machine Learning*.
Ed. by Jennifer Dy and Andreas Krause. Vol. 80.
Proceedings of Machine Learning Research. PMLR, 2018, pp. 2668–2677.
URL: https://proceedings.mlr.press/v80/kim18d.html (cit. on p. 37).

[77] Roberta Calegari, Giovanni Ciatto and Andrea Omicini.
“On the integration of symbolic and sub-symbolic techniques for XAI: A survey”.
In: *Intelligenza Artificiale* 14.1 (2020), pp. 7–32. URL: https://cris.unibo.it/retrieve/handle/11585/772707/677210/CCO-IA2019.pdf (cit. on pp. 37, 38, 58, 59, 61, 62, 68).

[78] Luciano Serafini and Artur S. d’Avila Garcez.
“Learning and Reasoning with Logic Tensor Networks”.
In: *AI∗IA 2016 Advances in Artificial Intelligence*. Springer International Publishing, 2016, pp. 334–348. DOI: 10.1007/978-3-319-49130-1_25 (cit. on p. 37).

[79] Tim Rocktäschel and Sebastian Riedel. “End-to-end Differentiable Proving”.
In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30.
Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/b2ab001909a8a6f04b51920306046ce5-Paper.pdf (cit. on p. 37).

[80] Roberto Confalonieri et al. “TREPAN Reloaded: A Knowledge-driven Approach to Explaining Artificial Neural Networks”. In: *ECAI 2020*. Frontiers in Artificial Intelligence and Applications 325 (2020), pp. 2457–2464. DOI: 10.3233/FAIA200378 (cit. on p. 37).

[81] Mark Craven and Jude Shavlik.
“Extracting Tree-Structured Representations of Trained Networks”.
In: *Advances in Neural Information Processing Systems*.
Ed. by D. Touretzky, M. C. Mozer and M. Hasselmo. Vol. 8. MIT Press, 1995.
URL: https://proceedings.neurips.cc/paper/1995/file/45f31d16b1058d586fc3be7207b58053-Paper.pdf (cit. on p. 37).

[82] Arne Seeliger, Matthias Pfaff and Helmut Krcmar. “Semantic web technologies for explainable machine learning models: A literature review.”
In: *PROFILES/SEMEX@ ISWC* 2465 (2019), pp. 1–16 (cit. on pp. 38, 82).

[83] Natasha Noy et al. “Industry-scale Knowledge Graphs: Lessons and Challenges”. In: *Queue* 17.2 (2019), pp. 48–75. DOI: 10.1145/3329781.3332266 (cit. on p. 38).

[84] *Wikidata*. URL: https://www.wikidata.org/wiki/Wikidata:Main_Page (visited on 16/03/2023) (cit. on p. 38).

[85] Robyn Speer, Joshua Chin and Catherine Havasi. “ConceptNet 5.5: An Open Multilingual Graph of General Knowledge”. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. Ed. by Satinder Singh and Shaul Markovitch. AAAI Press, 2017, pp. 4444–4451. URL: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972 (cit. on p. 38).

[86] Ilaria Tiddi and Stefan Schlobach. “Knowledge graphs as tools for explainable machine learning: A survey”. In: *Artificial Intelligence* 302 (2022), p. 103627. DOI: 10.1016/j.artint.2021.103627 (cit. on pp. 38, 92).

[87] Jose M. Alonso, Ciro Castiello and Corrado Mencar. “A Bibliometric Analysis of the Explainable Artificial Intelligence Research Field”. In: *Communications in Computer and Information Science*. Springer International Publishing, 2018, pp. 3–15. DOI: 10.1007/978-3-319-91473-2_1 (cit. on p. 38).

[88] Finale Doshi-Velez and Been Kim. “Considerations for Evaluation and Generalization in Interpretable Machine Learning”. In: *The Springer Series on Challenges in Machine Learning*. Springer International Publishing, 2018, pp. 3–17. DOI: 10.1007/978-3-319-98131-4_1 (cit. on p. 38).

[89] Ingrid Nunes and Dietmar Jannach. “A systematic review and taxonomy of explanations in decision support and recommender systems”. In: *User Modeling and User-Adapted Interaction* 27.3-5 (2017), pp. 393–444. DOI: 10.1007/s11257-017-9195-0 (cit. on p. 39).

[90] Diogo V. Carvalho, Eduardo M. Pereira and Jaime S. Cardoso. “Machine Learning Interpretability: A Survey on Methods and Metrics”. In: *Electronics* 8.8 (2019), p. 832. DOI: 10.3390/electronics8080832 (cit. on pp. 39, 41, 62, 63, 69).

[91] Jasper van der Waa et al. “Evaluating XAI: A comparison of rule-based and example-based explanations”. In: *Artificial Intelligence* 291 (2021), p. 103404. DOI: 10.1016/j.artint.2020.103404 (cit. on p. 39).

[92] Riccardo Guidotti. “Evaluating local explanation methods on ground truth”. In: *Artificial Intelligence* 291 (2021), p. 103428. DOI: 10.1016/j.artint.2020.103428 (cit. on p. 39).

[93] Leilani H. Gilpin et al. “Explaining Explanations: An Overview of Interpretability of Machine Learning”. In: *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2018. DOI: 10.1109/dsaa.2018.00018 (cit. on pp. 39, 58, 59).

[94] David Alvarez-Melis and Tommi S. Jaakkola. “On the Robustness of Interpretability Methods”. In: *WHI 2018*. 2018. URL: https://www.microsoft.com/en-us/research/publication/on-the-robustness-of-interpretability-methods/ (cit. on pp. 40, 43).

[95] Grégoire Montavon, Wojciech Samek and Klaus-Robert Müller.
"Methods for interpreting and understanding deep neural networks".
In: *Digital Signal Processing* 73 (2018), pp. 1–15. DOI: 10.1016/j.dsp.2017.10.011
(cit. on pp. 40, 43).

[96] Giulia Vilone and Luca Longo. "Explainable Artificial Intelligence: a Systematic Review".
In: (2020). DOI: 10.48550/ARXIV.2006.00093. arXiv: 2006.00093 [cs.AI]
(cit. on pp. 41, 43).

[97] Nancy Pennington and Reid Hastie.
"Explaining the evidence: Tests of the Story Model for juror decision making."
In: *Journal of Personality and Social Psychology* 62.2 (1992), pp. 189–206.
DOI: 10.1037/0022-3514.62.2.189 (cit. on p. 42).

[98] Michael Chromik and Martin Schuessler.
"A Taxonomy for Human Subject Evaluation of Black-Box Explanations in XAI."
In: *Exss-atec@ iui* 94 (2020). URL: https://ceur-ws.org/Vol-2582/paper9.pdf
(cit. on pp. 43, 45, 139).

[99] An-phi Nguyen and María Rodríguez Martínez.
"On quantitative aspects of model interpretability". In: (2020).
DOI: 10.48550/ARXIV.2007.07584. arXiv: 2007.07584 [cs.LG] (cit. on pp. 43, 45).

[100] Chirag Agarwal et al. "Rethinking Stability for Attribution-based Explanations".
In: *ICLR 2022 Workshop on PAIR^2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*. 2022.
URL: https://openreview.net/forum?id=BfxZAuWOg9 (cit. on p. 44).

[101] Mukund Sundararajan, Ankur Taly and Qiqi Yan.
"Axiomatic Attribution for Deep Networks".
In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*.
ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, pp. 3319–3328 (cit. on p. 44).

[102] Chih-Kuan Yeh et al. "On the (In)fidelity and Sensitivity of Explanations".
In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32.
Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/a7471fdc77b3435276507cc8f2dc2569-Paper.pdf (cit. on pp. 44, 63).

[103] Jose Oramas, Kaili Wang and Tinne Tuytelaars. "Visual Explanation by Interpretation: Improving Visual Feedback Capabilities of Deep Neural Networks".
In: *International Conference on Learning Representations*. 2019.
URL: https://openreview.net/forum?id=H1ziPjC5Fm (cit. on p. 44).

[104] Leila Arras et al.
""What is relevant in a text document?": An interpretable machine learning approach".
In: *PLOS ONE* 12.8 (2017). Ed. by Grigori Sidorov, e0181142.
DOI: 10.1371/journal.pone.0181142 (cit. on p. 44).

[105] Wojciech Samek et al.
"Evaluating the Visualization of What a Deep Neural Network Has Learned". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.11 (2017), pp. 2660–2673.
DOI: 10.1109/tnnls.2016.2599820 (cit. on p. 44).

[106] Martin Heusel et al.
"GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium".
In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30.
Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a1d694707eb0fefe65871369074926d-Paper.pdf (cit. on p. 45).

[107] Martin Pawelczyk, Klaus Broelemann and Gjergji Kasneci.
"Learning Model-Agnostic Counterfactual Explanations for Tabular Data".
In: *Proceedings of The Web Conference 2020*. ACM, 2020. DOI: 10.1145/3366423.3380087
(cit. on p. 45).

[108] Alex A. Freitas. "Comprehensible classification models".
In: *ACM SIGKDD Explorations Newsletter* 15.1 (2014), pp. 1–10.
DOI: 10.1145/2594473.2594475 (cit. on p. 45).

[109] Robert R. Hoffman et al. "Metrics for Explainable AI: Challenges and Prospects".
In: (2018). DOI: 10.48550/ARXIV.1812.04608. arXiv: 1812.04608 [cs.AI]
(cit. on pp. 45, 46, 139).

[110] Johan Huysmans et al. "An empirical evaluation of the comprehensibility of decision
table, tree and rule based predictive models".
In: *Decision Support Systems* 51.1 (2011), pp. 141–154.
DOI: 10.1016/j.dss.2010.12.003 (cit. on pp. 45–47).

[111] Jiun-Yin Jian, Ann M. Bisantz and Colin G. Drury.
"Foundations for an Empirically Determined Scale of Trust in Automated Systems".
In: *International Journal of Cognitive Ergonomics* 4.1 (2000), pp. 53–71.
DOI: 10.1207/s15327566ijce0401_04 (cit. on p. 46).

[112] Marta Garnelo and Murray Shanahan. "Reconciling deep learning with symbolic artificial
intelligence: representing objects and relations".
In: *Current Opinion in Behavioral Sciences* 29 (2019), pp. 17–23.
DOI: 10.1016/j.cobeha.2018.12.010 (cit. on p. 49).

[113] Stuart J. Russell and Peter Norvig.
Artificial Intelligence A Modern Approach. A Modern Approach. Pearson, 2020, p. 1136.
ISBN: 9780134610993 (cit. on p. 49).

[114] Stuart Charles Shapiro. *Encyclopedia of artificial intelligence*. Wiley, 1992, p. 1689.
ISBN: 047150307X (cit. on p. 49).

[115] Irving M. Copi, Carl Cohen and Victor Rodych. *Introduction to Logic*.
Taylor & Francis Group, 2018, p. 696. ISBN: 9781351386975 (cit. on p. 49).

[116] Bouke Kuijer. "Expressivity of Logics of Knowledge and Action". English.
PhD thesis. University of Groningen, 2014. ISBN: 978-90-367-7434-5 (cit. on p. 49).

[117] Alfred Tarski, Andrzej Mostowski and Raphael Mitchel Robinson. *Undecidable Theories*.
Ed. by Elsevier. Vol. 13. Studies in logic and the foundations of mathematics.
North-Holland, 1953. 98 pp. ISBN: 9780444533784.
URL: https://books.google.fr/books?id=XtLbjZjB1B8C (cit. on p. 49).

[118] Frank Van Harmelen, Vladimir Lifschitz and Bruce Porter.
Handbook of Knowledge Representation. Elsevier Science & Technology Books, 2008.
ISBN: 9780080557021 (cit. on pp. 49, 52).

[119] Tom Gruber. "Ontology". In: *Encyclopedia of Database Systems*. Springer New York, 2016,
pp. 1–3. DOI: 10.1007/978-1-4899-7993-3_1318-2 (cit. on p. 50).

[120] Karl Hammar. "Content Ontology Design Patterns: Qualities, Methods, and Tools".
PhD thesis. Linkoping University, 2017. DOI: 10.3384/diss.diva-139584 (cit. on p. 50).

[121] Mike Uschold and Michael Gruninger.
"Ontologies: principles, methods and applications".
In: *The Knowledge Engineering Review* 11.2 (1996), pp. 93–136.
DOI: 10.1017/s0269888900007797 (cit. on p. 50).

[122] R. Navigli, P. Velardi and A. Gangemi.
"Ontology learning and its application to automated terminology translation".
In: *IEEE Intelligent Systems* 18.1 (2003), pp. 22–31. DOI: 10.1109/mis.2003.1179190 (cit. on pp. 50, 51).

[123] Richard Arndt et al. "COMM: A Core Ontology for MultimediaAnnotation".
In: *Handbook on Ontologies*. Springer Berlin Heidelberg, 2009, pp. 403–421.
DOI: 10.1007/978-3-540-92673-3_18 (cit. on p. 50).

[124] Valentina Presutti and Aldo Gangemi.
"Content Ontology Design Patterns as Practical Building Blocks for Web Ontologies".
In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2008, pp. 128–141.
DOI: 10.1007/978-3-540-87877-3_11 (cit. on p. 50).

[125] W3C. *W3C SEMANTIC WEB ACTIVITY*. 2013.
URL: https://www.w3.org/2001/sw/ (visited on 07/04/2023) (cit. on p. 50).

[126] "Chapter Three - RDF and the Semantic Web Stack". In: *RDF Database Systems*.
Ed. by Olivier Curé and Guillaume Blin. Boston: Morgan Kaufmann, 2015, pp. 41–80.
ISBN: 978-0-12-799957-9.
DOI: https://doi.org/10.1016/B978-0-12-799957-9.00003-1. URL:
https://www.sciencedirect.com/science/article/pii/B9780127999579000031
(cit. on pp. 50, 51, 53, 54).

[127] Matthias Sesboüé et al.
"An Operational Architecture for Knowledge Graph-Based Systems".
In: *Procedia Computer Science* 207 (2022), pp. 1667–1676.
DOI: 10.1016/j.procs.2022.09.224 (cit. on p. 51).

[128] Jens Lehmann et al.
"DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia".
In: *Semantic Web* 6.2 (2015), pp. 167–195. DOI: 10.3233/sw-140134 (cit. on p. 51).

[129] Michael Ashburner et al. "Gene Ontology: tool for the unification of biology".
In: *Nature Genetics* 25.1 (2000), pp. 25–29. DOI: 10.1038/75556 (cit. on p. 51).

[130] Mike Bennett. "The financial industry business ontology: Best practice for big data".
In: *Journal of Banking Regulation* 14.3-4 (2013), pp. 255–268. DOI: 10.1057/jbr.2013.13
(cit. on p. 51).

[131] Freddy Brasileiro et al.
"Applying a Multi-Level Modeling Theory to Assess Taxonomic Hierarchies in Wikidata".
In: *Proceedings of the 25th International Conference Companion on World Wide Web - WWW '16 Companion*. ACM Press, 2016. DOI: 10.1145/2872518.2891117 (cit. on p. 51).

[132] Christiane Fellbaum.
WordNet. An Electronic Lexical Database (Language, Speech, and Communication).
The MIT Press, 1998, p. 423. ISBN: 9780262061971 (cit. on p. 52).

[133] Jia Deng et al. "ImageNet: A large-scale hierarchical image database".
In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.
DOI: 10.1109/cvpr.2009.5206848 (cit. on pp. 52, 131).

[134] Franz Baader et al., eds. *The Description Logic Handbook*.
Cambridge University Press, 2007. DOI: 10.1017/cbo9780511711787 (cit. on pp. 52, 53).

[135] Adila Krisnadhi and Pascal Hitzler. "Description Logics".
In: *Encyclopedia of Social Network Analysis and Mining*. Springer New York, 2018,
pp. 572–581. DOI: 10.1007/978-1-4939-7131-2_108 (cit. on pp. 52, 53).

[136] Sebastian Rudolph. "Foundations of Description Logics".
In: *Reasoning Web. Semantic Technologies for the Web of Data*.
Springer Berlin Heidelberg, 2011, pp. 76–136. DOI: 10.1007/978-3-642-23032-5_2.
eprint: https://www.aifb.kit.edu/images/1/19/DL-Intro.pdf (cit. on p. 53).

[137] W3C. *OWL Web Ontology Language Semantics and Abstract Syntax Section 5. RDF-Compatible Model-Theoretic Semantics*.
URL: https://www.w3.org/TR/owl-semantics/rdfs.html (visited on 21/09/2023)
(cit. on p. 53).

[138] W3C. *RDF 1.2 Schema*. 2023.
URL: https://www.w3.org/TR/rdf12-schema/ (visited on 21/09/2023) (cit. on p. 54).

[139] Ian Horrocks, Peter F. Patel-Schneider and Frank van Harmelen.
"From SHIQ and RDF to OWL: the making of a Web Ontology Language".
In: *Journal of Web Semantics* 1.1 (2003), pp. 7–26. DOI: 10.1016/j.websem.2003.07.001
(cit. on p. 54).

[140] Bernardo Cuenca Grau et al. "OWL 2: The next step for OWL".
In: *Journal of Web Semantics* 6.4 (2008), pp. 309–322.
DOI: 10.1016/j.websem.2008.05.001 (cit. on p. 54).

[141] W3C. *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax*.
2012. URL: https://www.w3.org/TR/owl2-syntax/ (visited on 15/03/2022)
(cit. on pp. 54, 56, 88, 92).

[142] W3C. *OWL 2 Web Ontology Language Mapping to RDF Graphs*. 2012.
URL: https://www.w3.org/TR/owl2-mapping-to-rdf/ (visited on 22/09/2023)
(cit. on p. 54).

[143] W3C. *OWL 2 Web Ontology Language Manchester Syntax*. 2012.
URL: https://www.w3.org/TR/owl2-manchester-syntax/ (visited on 15/03/2022)
(cit. on pp. 56, 85, 87).

[144] Mark A. Musen. "The protégé project: A Look Back and a Look Forward".
In: *AI Matters* 1.4 (2015), pp. 4–12. DOI: 10.1145/2757001.2757003 (cit. on pp. 56, 128).

[145] Jean-Baptiste Lamy. "Owlready: Ontology-oriented programming in Python with automatic classification and high level constructs for biomedical ontologies".
In: *Artificial Intelligence in Medicine* 80 (2017), pp. 11–28.
DOI: 10.1016/j.artmed.2017.07.002 (cit. on p. 56).

[146] Valérie Beaudouin et al.
"Flexible and Context-Specific AI Explainability: A Multidisciplinary Approach".
In: (2020). DOI: 10.48550/ARXIV.2003.07703. arXiv: 2003.07703 [cs.CY]
(cit. on pp. 58, 60, 61, 66).

[147] Giuseppe Futia and Antonio Vetrò. "On the Integration of Knowledge Graphs into Deep Learning Models for a More Comprehensible AI—Three Challenges for Future Research".
In: *Information* 11.2 (2020), p. 122. DOI: 10.3390/info11020122 (cit. on pp. 59–62, 82).

[148] Mike Wu et al. "Beyond Sparsity: Tree Regularization of Deep Models for Interpretability".
In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (2018).
DOI: 10.1609/aaai.v32i1.11501 (cit. on pp. 59, 60).

[149] *Artificial Intelligence in Society*. OECD, 2019. DOI: 10.1787/eedfee77-en (cit. on p. 60).

[150] Gary Klein. "Explaining Explanation, Part 3: The Causal Landscape".
In: *IEEE Intelligent Systems* 33.2 (2018), pp. 83–88. DOI: 10.1109/mis.2018.022441353
(cit. on p. 61).

[151] Robert Hoffman et al. "Explaining Explanation, Part 4: A Deep Dive on Deep Nets".
In: *IEEE Intelligent Systems* 33.3 (2018), pp. 87–95. DOI: 10.1109/mis.2018.033001421
(cit. on p. 61).

[152] Oana-Maria Camburu. “Explaining Deep Neural Networks”.
PhD thesis. Linacre College, University of Oxford, 2020. URL:
https://ora.ox.ac.uk/objects/uuid:2e9fa4f9-98c9-40af-a0d3-6c4203d46067
(cit. on p. 61).

[153] Amit Dhurandhar et al. “Explanations based on the Missing: Towards Contrastive Explanations with Pertinent Negatives”.
In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31.
Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/c5ff2543b53f4cc0ad3819a36752467b-Paper.pdf (cit. on p. 61).

[154] Aaron X. Fellmeth and Maurice Horwitz. *Guide to Latin in International Law*.
Oxford University Press, 2009. DOI: 10.1093/acref/9780195369380.001.0001
(cit. on p. 62).

[155] “ISO/IEC/IEEE International Standard - Systems and software engineering–Vocabulary”.
In: *ISO/IEC/IEEE 24765:2017(E)* (2017), pp. 1–541. DOI: 10.1109/IEEESTD.2017.8016712
(cit. on pp. 63, 67).

[156] Olivier Bousquet and André Elisseeff.
“Algorithmic Stability and Generalization Performance”.
In: *Advances in Neural Information Processing Systems*.
Ed. by T. Leen, T. Dietterich and V. Tresp. Vol. 13. MIT Press, 2000.
URL: https://proceedings.neurips.cc/paper_files/paper/2000/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf (cit. on p. 63).

[157] Heinrich Jiang et al. “To Trust Or Not To Trust A Classifier”.
In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio et al. Vol. 31.
Curran Associates, Inc., 2018. URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/7180cffd6a8e829dacfc2a31b3f72ece-Paper.pdf
(cit. on pp. 63, 77, 91, 137).

[158] Charles Corbière et al. “Addressing Failure Prediction by Learning Model Confidence”.
In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32.
Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/757f843a169cc678064d9530d12a1881-Paper.pdf
(cit. on pp. 63, 77).

[159] Marco Ancona et al. “Gradient-Based Attribution Methods”.
In: *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*.
Springer International Publishing, 2019, pp. 169–191.
DOI: 10.1007/978-3-030-28954-6_9 (cit. on p. 63).

[160] Janis Klaise et al. “Alibi Explain: Algorithms for Explaining Machine Learning Models”.
In: *Journal of Machine Learning Research* 22.181 (2021), pp. 1–7.
URL: http://jmlr.org/papers/v22/21-0017.html (cit. on p. 76).

[161] Harsha Nori et al.
“InterpretML: A Unified Framework for Machine Learning Interpretability”. In: (2019).
DOI: 10.48550/ARXIV.1909.09223. arXiv: 1909.09223 [cs.LG] (cit. on p. 76).

[162] Umang Bhatt et al. *Machine Learning Explainability for External Stakeholders*. 2020.
DOI: 10.48550/ARXIV.2007.05408 (cit. on p. 76).

[163] Larissa Chazette et al. “How Can We Develop Explainable Systems? Insights from a Literature Review and an Interview Study”.
In: *Proceedings of the International Conference on Software and System Processes and International Conference on Global Software Engineering*. ACM, 2022.
DOI: 10.1145/3529320.3529321 (cit. on p. 76).

[164] Dan Hendrycks and Kevin Gimpel. “A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks”. In: *International Conference on Learning Representations 2017* (2017). DOI: 10.48550/ARXIV.1610.02136. arXiv: 1610.02136 [cs.NE] (cit. on p. 77).

[165] Jiefeng Chen et al. “Detecting Errors and Estimating Accuracy on Unlabeled Data with Self-training Ensembles”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., 2021, pp. 14980–14992. URL: https://proceedings.neurips.cc/paper_files/paper/2021/file/7dd3ed2e12d7967b656d156d50308263-Paper.pdf (cit. on p. 77).

[166] Ching-Yao Chuang, Antonio Torralba and Stefanie Jegelka. “Estimating Generalization under Distribution Shifts via Domain-Invariant Representations”. In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by I. I. I. Hal Daumé and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 1984–1994. URL: https://proceedings.mlr.press/v119/chuang20a.html (cit. on p. 77).

[167] Xiaoge Zhang, Felix T. S. Chan and Sankaran Mahadevan. “Explainable machine learning in image classification models: An uncertainty quantification perspective”. In: *Knowledge-Based Systems* 243 (2022), p. 108418. DOI: 10.1016/j.knosys.2022.108418 (cit. on pp. 77, 78).

[168] Vitali Petsiuk, Abir Das and Kate Saenko. “RISE: Randomized Input Sampling for Explanation of Black-box Models”. In: (2018), p. 151. URL: http://bmvc2018.org/contents/papers/1064.pdf (cit. on p. 78).

[169] Bolei Zhou et al. “Object Detectors Emerge in Deep Scene CNNs”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1412.6856 (cit. on p. 78).

[170] Piotr Dabkowski and Yarin Gal. “Real Time Image Saliency for Black Box Classifiers”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/0060ef47b12160b9198302ebdb144dcf-Paper.pdf (cit. on p. 78).

[171] J. T. Springenberg et al. “Striving for Simplicity: The All Convolutional Net”. In: *ICLR (workshop track)*. 2015. URL: http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a (cit. on p. 79).

[172] Jianming Zhang et al. “Top-Down Neural Attention by Excitation Backprop”. In: *International Journal of Computer Vision* 126.10 (2017), pp. 1084–1102. DOI: 10.1007/s11263-017-1059-x (cit. on p. 79).

[173] Bolei Zhou et al. “Learning Deep Features for Discriminative Localization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 79).

[174] 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 (ICCV)*. 2017 (cit. on p. 79).

[175] Anna Nguyen, Adrian Oberföll and Michael Färber. “Right for the Right Reasons: Making Image Classification Intuitively Explainable”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2021, pp. 327–333. DOI: 10.1007/978-3-030-72240-1_32 (cit. on p. 79).

[176] Tom Vermeire et al. “Explainable image classification with evidence counterfactual”. In: *Pattern Analysis and Applications* 25.2 (2022), pp. 315–335. DOI: 10.1007/s10044-021-01055-y (cit. on pp. 79, 80, 96).

[177] Lisa Anne Hendricks et al. “Generating Visual Explanations”. In: *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 3–19. DOI: 10.1007/978-3-319-46493-0_1 (cit. on p. 80).

[178] Lisa Anne Hendricks et al. “Generating visual explanations with natural language”. In: *Applied AI Letters* 2.4 (2021). DOI: 10.1002/ail2.55 (cit. on pp. 80, 82).

[179] Frank van Harmelen and Annette ten Teije. “A Boxology of Design Patterns for Hybrid Learning and Reasoning Systems”. In: *Journal of Web Engineering* 18.3 (2019), pp. 97–124. DOI: 10.13052/jwe1540-9589.18133 (cit. on pp. 82, 86, 93).

[180] Nhathai Phan et al. “Ontology-based deep learning for human behavior prediction with explanations in health social networks”. In: *Information Sciences* 384 (2017), pp. 298–313. DOI: 10.1016/j.ins.2016.08.038 (cit. on p. 82).

[181] Jeroen Voogd et al. “Using Relational Concept Networks for Explainable Decision Support”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2019, pp. 78–93. DOI: 10.1007/978-3-030-29726-8_6 (cit. on pp. 82, 83, 86).

[182] Grégory Bourguin et al. “Towards Ontologically Explainable Classifiers”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2021, pp. 472–484. DOI: 10.1007/978-3-030-86340-1_38 (cit. on pp. 83, 86, 92).

[183] Marjan Alirezaie et al. “A symbolic approach for explaining errors in image classification tasks”. In: *Working Papers and Documents of the IJCAI-ECAI-2018 Workshop on*. 2018. URL: https://msioutis.gitlab.io/files/lr2018.pdf (cit. on p. 83).

[184] Q. Vera Liao, Daniel Gruen and Sarah Miller. “Questioning the AI: Informing Design Practices for Explainable AI User Experiences”. In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, 2020. DOI: 10.1145/3313831.3376590 (cit. on pp. 83, 84).

[185] Michael Chromik and Andreas Butz. “Human-XAI Interaction: A Review and Design Principles for Explanation User Interfaces”. In: *Human-Computer Interaction – INTERACT 2021*. Springer International Publishing, 2021, pp. 619–640. DOI: 10.1007/978-3-030-85616-8_36 (cit. on pp. 83, 84, 94, 97).

[186] Saleema Amershi et al. “Guidelines for Human-AI Interaction”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 2019. DOI: 10.1145/3290605.3300233 (cit. on p. 84).

[187] James Wexler et al. “The What-If Tool: Interactive Probing of Machine Learning Models”. In: *IEEE Transactions on Visualization and Computer Graphics* (2019), pp. 1–1. DOI: 10.1109/tvcg.2019.2934619 (cit. on p. 84).

[188] Zoe Zhang. “User Interface Design Based on Human-Centered XAI Methods”. MA thesis. University of Twente, 2022. URL: http://essay.utwente.nl/93900/ (cit. on p. 84).

[189] João Rafael Gomes Varela. “Interface Design for Human-guided Explainable AI”. MA thesis. Faculty of Engineering of the University of Porto, 2022 (cit. on p. 84).

[190] Hubert Baniecki et al. “dalex: Responsible Machine Learning with Interactive Explainability and Fairness in Python”.
In: *Journal of Machine Learning Research* 22.214 (2021), pp. 1–7.
URL: http://jmlr.org/papers/v22/20-1473.html (cit. on p. 84).

[191] Weina Jin et al. “EUCA: the End-User-Centered Explainable AI Framework”. In: (2021).
DOI: 10.48550/ARXIV.2102.02437. arXiv: 2102.02437 [cs.HC] (cit. on p. 84).

[192] Neal J. Roese. “Counterfactual thinking.”
In: *Psychological Bulletin* 121.1 (1997), pp. 133–148.
DOI: 10.1037/0033-2909.121.1.133 (cit. on p. 100).

[193] Riccardo Guidotti. “Counterfactual explanations and how to find them: literature review and benchmarking”. In: *Data Mining and Knowledge Discovery* (2022).
DOI: 10.1007/s10618-022-00831-6 (cit. on pp. 100–105, 109).

[194] Peter Lipton. “Contrastive Explanation”.
In: *Royal Institute of Philosophy Supplement* 27 (1990), pp. 247–266.
DOI: 10.1017/s1358246100005130 (cit. on p. 101).

[195] Tim Miller. “Contrastive explanation: a structural-model approach”.
In: *The Knowledge Engineering Review* 36 (2021). DOI: 10.1017/s0269888921000102 (cit. on pp. 101, 102).

[196] Mark T. Keane and Barry Smyth. “Good Counterfactuals and Where to Find Them: A Case-Based Technique for Generating Counterfactuals for Explainable AI (XAI)”.
In: *Case-Based Reasoning Research and Development*.
Springer International Publishing, 2020, pp. 163–178.
DOI: 10.1007/978-3-030-58342-2_11.
eprint: https://arxiv.org/ftp/arxiv/papers/2005/2005.13997.pdf
(cit. on pp. 101, 103).

[197] Ruth M. J. Byrne. “Counterfactuals in Explainable Artificial Intelligence (XAI): Evidence from Human Reasoning.” In: *IJCAI 2019*. 2019, pp. 6276–628 (cit. on p. 101).

[198] Jonathan Dodge et al. “Explaining models”.
In: *Proceedings of the 24th International Conference on Intelligent User Interfaces*.
ACM, 2019. DOI: 10.1145/3301275.3302310 (cit. on p. 101).

[199] Reuben Binns et al. “’It’s Reducing a Human Being to a Percentage’”.
In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*.
ACM, 2018. DOI: 10.1145/3173574.3173951 (cit. on p. 101).

[200] Adam White and Artur S. d’Avila Garcez.
“Measurable Counterfactual Local Explanations for Any Classifier”.
In: *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*.
Ed. by Giuseppe De Giacomo et al. Vol. 325.
Frontiers in Artificial Intelligence and Applications. IOS Press, 2020, pp. 2529–2535.
DOI: 10.3233/FAIA200387. URL: https://doi.org/10.3233/FAIA200387
(cit. on pp. 102, 103).

[201] Amir-Hossein Karimi et al.
“Model-Agnostic Counterfactual Explanations for Consequential Decisions”.
In: *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*. Ed. by Silvia Chiappa and Roberto Calandra. Vol. 108.
Proceedings of Machine Learning Research. PMLR, 2020, pp. 895–905.
URL: https://proceedings.mlr.press/v108/karimi20a.html (cit. on pp. 102, 103).

[202] Maximilian Schleich et al. “GeCo: Quality Counterfactual Explanations in Real Time”. In: *Proceedings of the VLDB Endowment* 14.9 (2021), pp. 1681–1693. DOI: 10.14778/3461535.3461555. arXiv: 2101.01292 [cs.LG] (cit. on pp. 103, 105, 109, 110).

[203] Rafael Poyiadzi et al. “FACE”. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. ACM, 2020. DOI: 10.1145/3375627.3375850 (cit. on p. 103).

[204] Shubham Sharma, Jette Henderson and Joydeep Ghosh. “CERTIFAI: Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models”. In: (2019). DOI: 10.1145/3375627.3375812. arXiv: 1905.07857 [cs.LG] (cit. on pp. 104, 106).

[205] Matthew Horridge. “Justification based explanation in ontologies”. PhD thesis. 2011 (cit. on p. 106).

[206] Aditya Kalyanpur et al. “Debugging unsatisfiable classes in OWL ontologies”. In: *Journal of Web Semantics* 3.4 (2005), pp. 268–293. DOI: 10.1016/j.websem.2005.09.005 (cit. on p. 106).

[207] Konstantin Schekotihin, Patrick Rodler and Wolfgang Schmid. “OntoDebug: Interactive Ontology Debugging Plug-in for Protégé”. In: *Lecture Notes in Computer Science*. Springer International Publishing, 2018, pp. 340–359. DOI: 10.1007/978-3-319-90050-6_19 (cit. on p. 106).

[208] Christian Alrabbaa et al. “Finding Small Proofs for Description Logic Entailments: Theory and Practice (Extended Technical Report)”. In: *LPAR-23: 23rd International Conference on Logic for Programming, Artificial Intelligence and Reasoning, vol 73, 2020, pages 32–67* (2020). DOI: 10.29007/nhpp. arXiv: 2004.08311 [cs.LO] (cit. on p. 106).

[209] Patrick Lambrix. “Completing and Debugging Ontologies: state of the art and challenges”. In: (2019). arXiv: 1908.03171 [cs.AI] (cit. on p. 106).

[210] Christian Alrabbaa et al. “On the eve of true explainability for OWL ontologies: Description logic proofs with Evee and Evonne”. In: *Proc. DL* 22 (2022) (cit. on p. 106).

[211] Simone Coetzer and Katarina Britz. “Debugging Classical Ontologies Using Defeasible Reasoning Tools”. In: *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2021. DOI: 10.3233/faia210374 (cit. on p. 106).

[212] Markel Vigo et al. “Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design”. In: *International Journal of Human-Computer Studies* 72.12 (2014), pp. 835–845. DOI: 10.1016/j.ijhcs.2014.07.005 (cit. on p. 106).

[213] Jérôme Euzenat et al. *Ontology distances for contextualisation*. Contract. euzenat2009b. INRIA, 2009, p. 50. URL: https://hal.inria.fr/hal-00793450 (cit. on pp. 106–108).

[214] Santiago Ontañón. “An overview of distance and similarity functions for structured data”. In: *Artificial Intelligence Review* 53.7 (2020), pp. 5309–5351. DOI: 10.1007/s10462-020-09821-w (cit. on pp. 107, 108).

[215] Bo Hu et al. “Semantic Metrics”. In: *Managing Knowledge in a World of Networks*. Springer Berlin Heidelberg, 2006, pp. 166–181. DOI: 10.1007/11891451_17 (cit. on pp. 107, 108).

[216] Carmen Fernández-Chamizo et al. “Supporting object reuse through case-based reasoning”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 1996, pp. 135–149. DOI: 10.1007/bfb0020607 (cit. on pp. 107, 108, 121).

[217] Krzysztof Janowicz. “Sim-DL: Towards a Semantic Similarity Measurement Theory for the Description Logic ALCNR in Geographic Information Retrieval”.
In: *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*.
Springer Berlin Heidelberg, 2006, pp. 1681–1692. DOI: 10.1007/11915072_74
(cit. on p. 108).

[218] Mirna El Ghosh et al. “Rel Topic : A graph-based semantic relatedness measure in topic ontologies and its applicability for topic labeling of old press articles”.
In: *Semantic Web* 14.2 (2022). Ed. by Mehwish Alam et al., pp. 293–321.
DOI: 10.3233/sw-222919 (cit. on p. 108).

[219] R. Rada et al. “Development and application of a metric on semantic nets”.
In: *IEEE Transactions on Systems, Man, and Cybernetics* 19.1 (1989), pp. 17–30.
DOI: 10.1109/21.24528 (cit. on pp. 108, 122).

[220] P. Resnik. “Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language”.
In: *Journal of Artificial Intelligence Research* 11 (1999), pp. 95–130.
DOI: 10.1613/jair.514 (cit. on p. 108).

[221] Xinbo Gao et al. “A survey of graph edit distance”.
In: *Pattern Analysis and Applications* 13.1 (2009), pp. 113–129.
DOI: 10.1007/s10044-008-0141-y (cit. on p. 108).

[222] Gary Chartrand. *Introductory graph theory*. Dover, 1985, p. 294. ISBN: 0486247759
(cit. on pp. 114, 121).

[223] Erich M. von Hornbostel and Curt Sachs. “Classification of Musical Instruments: Translated from the Original German by Anthony Baines and Klaus P. Wachsmann”.
In: *The Galpin Society Journal* 14 (1961), p. 3. DOI: 10.2307/842168
(cit. on pp. 128, 129, 153, 154).

[224] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 131).

[225] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton.
“ImageNet Classification with Deep Convolutional Neural Networks”.
In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25.
Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf (cit. on p. 131).

[226] Karen Simonyan and Andrew Zisserman.
“Very Deep Convolutional Networks for Large-Scale Image Recognition”.
In: *International Conference on Learning Representations*. 2015 (cit. on p. 131).

[227] Gao Huang et al. “Convolutional Networks with Dense Connectivity”.
In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019) (cit. on p. 131).

[228] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”.
In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016.
DOI: 10.1109/cvpr.2016.308 (cit. on p. 131).

[229] Massih-Reza Amini and Éric Gaussier.
Recherche d’Information - applications, modèles et algorithmes. Eyrolles, 2013.
ISBN: 978-2-212-13532-9.
URL: http://www.eyrolles.com/Informatique/Livre/recherche-d-information-9782212135329 (cit. on p. 140).