UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA
DIPARTIMENTO DI INFORMATICA, SISTEMISTICA E COMUNICAZIONE
CORSO DI DOTTORATO IN COMPUTER SCIENCE

Empowering XAI and LMI with Human-in-the-loop

Supervisor: Prof. Mario Mezzanzanica
Co-supervisor: Prof. Fabio Mercorio
Tutor: Prof. Gabriella Pasi

Candidate:
NAVID NOBANI
NUM. 836807

Academic Year 2021/2022

"Study hard what interests you the most in the most undisciplined, irreverent and original manner possible."

Richard P. Feynman

Abstract

Once aimed to mimic the human brain and existed only as mathematical models in academia, the vast family of Artificial Intelligence methods are well passed that initial goal; models with billions of parameters trained on millions of mostly human-generated data. Such models are present in almost each and every aspect of our lives, from the weather forecasts and social network content to how our banks detect fraudulent transactions connected to our accounts and maps that guide us to a restaurant through unknown streets of a new city.

All these advancements, though, have a common limitation: their performance is bounded to the amount of human knowledge we can feed them, i.e. training data that should chase the ever-growing model parameters both in terms of quantity and quality. Unfortunately, such a requirement often comes with a high cost, given that generating machine-friendly data and updating and maintaining them is enormously labour-intensive.

One way to overcome this issue is the Human-in-the-Loop (HITL) paradigm: looking at humans not only as a passive part of the system, i.e. provider of inputs and consumer of outputs but as an active part of AI systems that participate in the creation and validation of data, model parameters and model inputs. By doing so, we inject the system with up-to-date human knowledge that otherwise should have arrived through expensive and often outdated training sets.

This thesis proposes novel methods for integrating HITL with the eXplainable Artificial Intelligence (XAI) and Labour Market Intelligence (LMI) fields:

In part I, We propose and implement a conversational explanation system called `ConvXAI` by extending the current state-of-the-art and introducing a new conversation type, i.e. Clarification conversation. Following the HITL paradigm, `ConvXAI` differentiates itself from the classic XAI systems that create one-size-fit-all explanations regardless of the user's knowledge level, background and need by providing explanations that fit the user's context

and using the information provided by the user. This model is made by anonymous data provided by Digital Attitude S.r.l company.

In part II, we provide a model called `TaxoRef`, which achieves its objective, i.e. taxonomy refinement, by considering domain experts as providers of the input data (taxonomy) and in the same time, as final validators of the model's suggestions. This method was developed by data provided by Tabulaex/Burning Glass Technologies company.

Acknowledgments

First and foremost, I want to express my sincere gratitude to Prof. Mario Mezzanzanica and Prof. Fabio Mercorio, my PhD supervisors, for their wise counsel, unwavering encouragement, and tolerance. During my academic research, their vast knowledge and wealth of experience have inspired me. I also want to thank Prof. Gabriella Pasi for her technical support of my study. I would like to thank all my former and current colleagues in TabulaeX, CRISP and Digital Attitude. It is their kind help and support that help me during my PhD years.

I also thank my uncle, Dr. Ahmad Heidari who helped me in each and every step of my life. Finally, I would like to express my gratitude to my parents, who encouraged me to keep pushing forward regardless of the obstacles. I am sorry that my dad cannot see me graduate.

Navid

Contents

Abstract v

Acknowledgements vii

I HITL in XAI 1

1 Introduction and Background 3

1.1 The Need for Human-in-the-loop Approach 3

1.1.1 HITL in XAI 5

1.1.2 HITL in LMI 6

1.2 Explainable AI 7

1.3 Conversational Systems 12

1.4 Thesis Structure 16

2 Natural Language Explanations 17

2.1 Does XAI need Natural Language Explanations? 19

2.1.1 A Roadmap for Selecting XAI-based Systems 19

2.2 Context Definition 21

2.2.1 Explanation Goal 21

2.2.2 Audience 21

2.3 Explanation Generation 23

2.3.1 Explanator Type 24

2.3.2 Structure 24

2.3.3 Explanation Type 25

2.4 Message Generation 27
2.4.1 Presentation Technique 27
2.5 Keeping the roadmap up-to-date 38
2.5.1 Multi-Criteria Decision-Making at a glance 39

3 Explaining black box Classifiers Through Natural Language 45
3.1 ContrXT in a Nutshell 46
3.2 Results on a Benchmark Dataset 49

4 Conversational Explanations 53
4.0.1 Motivating Example. 55
4.0.2 Contribution. 56
4.1 Problem Formulation 56
4.2 How ConvXAI works 60
4.2.1 Conversation Initialiser 62
4.2.2 Natural Language Understanding 63
4.2.3 Dialogue State Tracking 67
4.2.4 Dialogue Policy 68
4.2.5 Explanation Generator 69
4.2.6 Schema 71
4.2.7 Summary of ConvXAI tool 71
4.2.8 Tool and User Interface 71
4.3 Evaluating ConvXAI through a User Study 75
4.3.1 Research Method 75
4.3.2 Experimental Design 76
4.3.3 Result Comments 77
4.4 Conclusion 90

II HITL in LMI 91

5 Setting the stage on LMI 93
5.1 The Significance of Analysing Job Ads 95

5.2 Preliminaries and Related Work 96
5.2.1 Word Embeddings 97
5.2.2 Taxonomies 100

6 Embedding Evaluation Through Semantic Similarity 105
6.1 Methodology 107
6.1.1 Step 1: Hierarchical Semantic Similarity (HSS) 107
6.1.2 Step 2: A Tool for Computing Semantic Similarity 109
6.1.3 Step 3: Embeddings Selection and Evaluation of the Best Embedding 109
6.2 Experimental Results 111
6.2.1 Step 1: Evaluation of the HSS 112
6.2.2 Step 2: A Tool for Computing Semantic Similarity 115
6.2.3 Step 3: Embeddings Selection and Evaluation of the Best Embedding 115

7 Taxonomy Refinement Through Embedding Evaluation 125
7.1 The TaxoRef Approach 126
7.2 Experimental Results on 2M+ UK Online Job Ads 127
7.2.1 Result Comments 129

8 Conclusion 135
8.1 Future Works 136

9 Acronyms 137

List of Figures 141

List of Tables 145

Bibliography 147

Part I

HITL in XAI

1

Introduction and Background

1.1 The Need for Human-in-the-loop Approach

Machine Learning (ML) models, especially a particular group of them called Deep Learning (DL) models, are rapidly gaining popularity among researchers and popular culture, turning what once was considered science fiction (e.g. universal translators see [2]) into science. Regardless of the hype and the actual achievements of DL models, improving them and pushing the current frontiers faces a serious blocker: training data or, more specifically, the disproportionate growth rates of model parameters and features from one side and the quantity of the training records in hand from another hand.

For instance, let us consider a model which predicts when supermarket customers deviate from their current behaviour traits. Thanks to the increasing number of touchpoints customers interact with (Their position and movement velocity while shopping, products they examine but not buy, products they buy together...the list goes on), we can come up with new features to feed into the algorithm hoping for a better performing model. But regardless of our idea about the potential improvements these newly introduced features bring, the number of

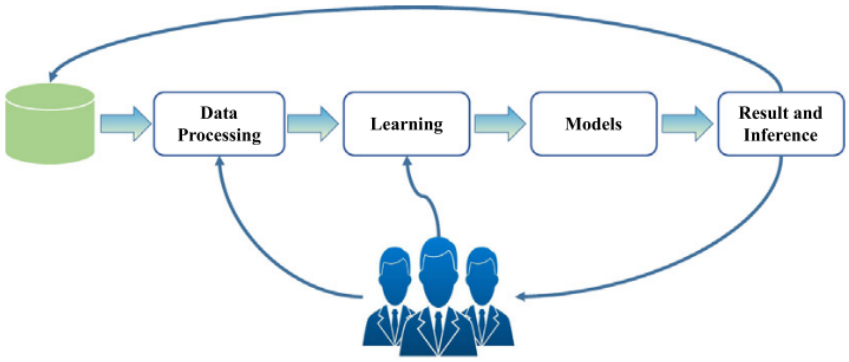


Figure 1.1 The development cycle of model [303]

records (i.e. number of surviving customers) remains the same. Such incoherency between the number of features and records will drastically impact the success of the generated model.

To minimize this issue, researchers introduced various solutions like Automatic generation of datasets (see e.g. [136, 75]), Transfer learning (see e.g. [216, 297]) and one-shot and few-shot learning (see e.g. [83, 267]). Another approach to tackling this problem is incorporating pre-training knowledge into the learning frameworks [303]. To this end, researchers started to look at humans as a source of wisdom and knowledge which could overcome the problem of scarce data by integrating human knowledge in machine learning systems both in the data processing and learning phases. Figure 1.1 shows a conventional ML algorithm combined with human agents [303].

The mentioned studies contribute to the born of an approach called "Human-in-the-Loop" (HITL) which, broadly speaking, tries to integrate human knowledge into machine learning systems to reach results that the algorithm could not have achieved.

It is worth mentioning that even though DL models benefit substantially from HITL methods, HITL pre-dates DL models by decades. The literature includes examples from aerospace and maritime applications (see, e.g. [220, 191]) to medicine and music industry (see e.g. [41, 124, 40])

The recent survey on HITL applications in ML by wu et al. [303] divides HITL applications into three categories called Data processing, Model training and inference and System construction and application.

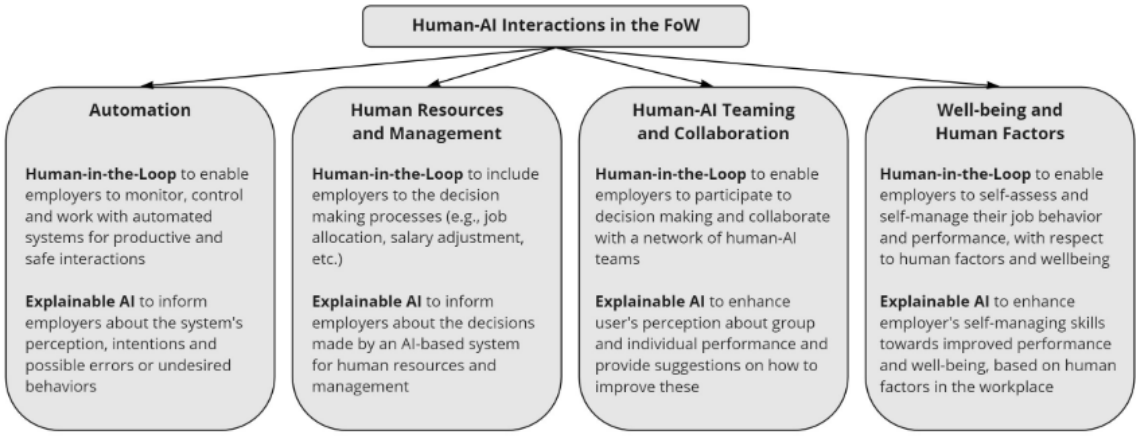


Figure 1.2 Human-in-the-Loop and Explainable AI in the Future of Work [272]

1.1.1 HITL in XAI

Similar to HITL, XAI methods were born to enhance the role and impact of humans in AI systems following human-centric design principals. Such a similarity often permits researchers to integrate HITL and XAI together in order to amplify the role of humans in both creating and interpreting black boxes. In their work Tsiakas et al. [272] study how HITL and XAI can be integrated together towards the Future of Work (FoW), arguing that the combination of these approaches can create new design possibilities in human-AI interaction and the FoW. They further identify four contexts in which HITL and XAI can be integrated (see Figure 1.2).

In Part I of this thesis, we study the application of HITL within an XAI and discuss how such integration benefits the system as a whole and, above all, the end users. We achieve such a goal by providing a conversational system that enables users to enhance their trust in the AI system upon receiving tailored explanations based on their personal preferences. Such explanations can be generated around What, Why, Why Not, How, and What If questions. On top of it, users are able to control how explanations are created both from contextual (e.g. using specific semantics for technical users) and presentation (e.g. textual vs graphical explanations) points of view.

1.1.2 HITL in LMI

The volume, variety and velocity of labour market data continue to increase. Vast amounts of digital data are generated by people in various channels like social media and by organisations through their internal and external networks and interaction with these parties. The endeavour to make sense of these data brings about exciting opportunities.

Colace et al. [62] define Labour Market Intelligence or LMI as "The definition of AI algorithms and frameworks that derive useful knowledge for labour market-related activities, by putting AI into the labour market". Such a definition will open a vast horizon of methods, approaches and applications like extracting needed skills for various occupations, discrepancy detection in labour skills, identification of obsolete skills and demand prediction for new emerging occupations. Having a strong bond with Natural Language Processing (NLP) field, most analyses done in LMI are based on Lexical taxonomies and distributional representations, e.g. , semantic similarity measurements and taxonomy learning. Recently, several scholars have proposed new approaches to combine those resources into a unified representation preserving distributional and knowledge-based lexical features.

Lexical taxonomies are a natural method for organising human knowledge in a hierarchical form and provide a formal description of concepts and their relations and support syntactic and semantic exchanges. Contextually, word embeddings have gained remarkable popularity in computational linguistics, mostly thanks to their ability to extract linguistic patterns and lexical semantics from large corpora. However, despite their wide usage, finding a unified measure accounting for both knowledge-based and distributional resources is still an open problem.

Word embeddings assume that words occurring in the same context tend to have similar meanings. These methods are semi-supervised and knowledge-poor, thus suitable for large corpora and evolving scenarios.

Taxonomies are a natural way to represent and organise concepts in a hierarchical manner. They are pivotal for machine understanding, natural language processing, and decision-making tasks. However, taxonomies are domain-dependent, usually have low coverage, and their manual creation and updating are time-consuming and require domain-specific knowledge [87]. For these reasons, many researchers have tried to automatically infer

semantic information from domain-specific text corpora to build or update taxonomies. Despite automated construction of new taxonomies from scratch bieng a well-established research area [291], the refinement of existing hierarchies is far from being considered as a mature field. Due to the evolution of human languages and the proliferation of online content, it is often required to improve existing taxonomies while maintaining their structure. To date, the most adopted approaches to enrich or extend standard *de-jure* taxonomies lean on expert panels and surveys.

Following we list our published on the LMI subject:

- Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., & **Nobani, N**. "Embeddings Evaluation Using a Novel Measure of Semantic Similarity." Cognitive Computation 14.2 (2022): 749-763.
- **Nobani, N.**, Malandri, L., Mercorio, F., & Mezzanzanica, M. "A Method for Taxonomy-Aware Embeddings Evaluation (Student Abstract)." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 18. 2021.
- Malandri, L., Mercorio, F., Mezzanzanica, M., & **Nobani, N.** "TaxoRef: Embeddings Evaluation for AI-driven Taxonomy Refinement." Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, 2021.
- Malandri, L., Mercorio, F., Mezzanzanica, M., & **Nobani, N.** "MEET-LM: A method for embeddings evaluation for taxonomic data in the labour market." Computers in Industry 124 (2021): 103341.
- Malandri, L., Mercorio, F., Mezzanzanica, M., & **Nobani, N.** "Meet: A method for embeddings evaluation for taxonomic data." 2020 International Conference on Data Mining Workshops (ICDMW). IEEE, 2020.

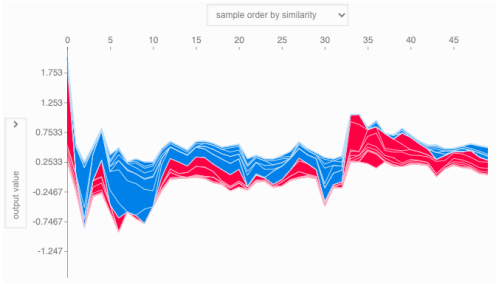
1.2 Explainable AI

XAI is becoming an indispensable component of AI-based human-centric systems in several fields and industries, ranging from health care [219] and legal applications [71] to robotics [88]. Such methods are utilised to offer explanations and interpretations of how black boxes work internally and how their decisions are made, as clarified by [106]. While a few years ago such features were considered *"nice to have"*, the recent proliferation of

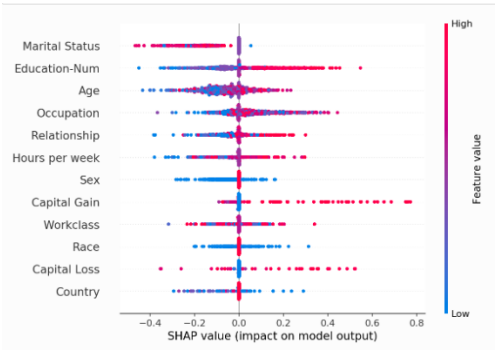
potent but opaque models on one side and the force from the legislation on the other side are bringing XAI more and more towards being a commodity. For instance, the GDPR (General Data Protection Regulation), adopted by the European Union in 2018, defines the *right to the explanations* (Art. 13-15), asking the data controllers to provide data subjects with information about *"the existence of automated decision-making, including profiling, referred to in Article 22(1) and (4) and, at least in those cases, meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject"*. The focus is on the adjective *meaningful*, as this implies that any stakeholder should be in a state to understand the logic behind an automated process without any prior technical knowledge. This, as a consequence, attributes to the natural language a key role in providing explanations to end-users in an effective way. Despite the prominence of XAI methods in recent AI literature and the ever-widening range of their application domains, the attention paid to the *"last mile"* of the XAI-based systems,i.e. the presentation of explanations to end-users, is still in a growing phase.

This could lead to solutions that, while potent and practical from a technical point of view, cannot be directly utilised by non-technical users, defying the principal objective of an XAI system (see [204]). For instance, we can mention LIME [238] and SHAP [179] as two of the most important and influential XAI methods despite having a considerable impact on the XAI field, produce explanations that are often difficult or impossible to interpret by a layperson. Figure 1.3 provides some examples of explanations generated by these methods.

Such an issue often arises by neglecting the presentation method that is used to convey the explanations to the users. An example of a human-centred XAI that considers both the context and the presentation model is proposed by [249]. Different presentation methods (i.e. how explanations are conveyed to users) exist in the literature.; for example, graphics/plots [177], images [239], reports [146] and Natural Language [193]). While the presentation method is not the only choice to be made when implementing an XAI system, it has a direct effect on user comprehension and, therefore, on the success of the explanation, as pointed out by [131], where the authors discuss how different types of explanations could affect the success of the XAI model in general. Therefore, we argue that it should be the user to guides the explanation and presentation process.

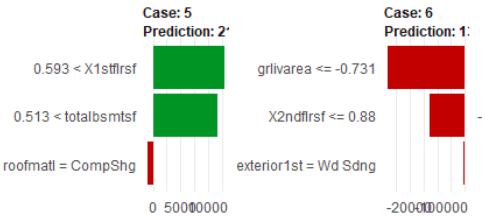


(a) LIME: Force plot



(b) SHAP: Summary plot

(c) LIME: Explaining prediction of 'Cat'



(d) LIME: Feature explanation

Figure 1.3 Example of explanations provided by SHAP and LIME explainers

In a user study conducted by [90], the authors examined ten different explanation types and measured their effects in different dimensions, concluding that certain types of graphical explanations (tag clouds) have increased the level of transparency perceived by the users and, consequently, their satisfaction.

Among different methods of presentation, those using Natural Language (NL) can effectively improve decision-making under uncertainty with respect to other presentation methods, such as graphical presentations, as claimed by [99] based on a task-based study with 442 adults. Furthermore, NL explanations are also easier to comprehend by non-technical users, as pointed out by [193]. A subfield of natural language processing (NLP) is *conversational systems*, which enables users to argue with the system to obtain their desired explanations. While in recent years, several works have proposed models and frameworks regarding such methods, relative to other NLP methods, dialogical methods are less explored, and lack of empirical validation makes it difficult to assess their effectiveness in real-life applications, as discussed by [182].

Explanations made through Natural Language, as pointed out by [193] are the key component of the future intelligent interactive agents, given their ability to offer interpretability to people with a diverse background and to better mimic humans, which usually explain their decision verbally. Moreover, in [99], the authors mention that the use of NL improves decision-making under uncertainty compared to graphical-based presentation methods, a claim which is supported by the work of [89], showing that NL presentations are comparable in effectiveness for decision support to the other forms of presentations.

Using NL methods in explanation creation has advantages like higher efficiency and coverage in terms of audience. Another element engaged, is the power of these presentation techniques to convey social cues (see, e.g. [54]), interacting with users and reinforcing their trust in the information system as a whole beyond the black box model and its underneath data (see, e.g. [70]).

In [260] the authors argue that NL explanations are suitable for lay audiences given that their interaction mode gives the process a natural feel, while [151] make a step further by observing that NL presentations increase the acceptance and trust levels in end-users. These explanations are more efficient (see [11]), more insightful for specific cases concerning the visual methods (see [218]) and target a broader range of users. Furthermore, [193] argues

that these explanations leverage the common language with the user, profiting from the mental concepts which are already established in the human language (see [236]). [152] argues that generating a text that mimics how humans use natural language to explain, describe, or inform is not a straightforward task, nor to choose the message communicated or transforming it into natural language. Such potential benefits can be achieved when XAI solutions evolve from static, one-directional messages and go towards dialogues that directly engage the end-user in the explanation process by offering rich and personalised interactions that mimic how humans explain their decisions. [257] formulates this need as "a natural pairing between Explainable AI and Argumentation while the first requires the need to clarify and defend decisions and the second provides a method for linking any decision to the evidence supporting it". On the other hand, focusing on the performance of such explanations, [182] state that interactive explanations can "*provide richer and satisfactory explanations as opposed to one-shot explanations"*.

To address this need, several models have been proposed for conversational explanations based on the field of computational argumentation(see [68] for a review on this field). In [287, 288], Walton introduces and formalises the *dialectical explanatory dialogue*. This system consists of three components which are opening and closing moves, speech act and a series of rules to govern the speech acts and evaluate the success of the conversation. [18] extends Walton's model by adding the concept of "dialectical shifts". [182] use the grounding theory [101] to propose agent dialogue framework (ADF) and argue that *"people switch from explanation to argumentation and back again during an explanation dialogue"*. A limited number of works in the field of XAI propose their explanations through dialogues and interactive natural language presentations. [261] propose a system that demonstrates a system that explains predictions with class-contrastive counterfactual statements through a voice-enabled dialogue. [155] build a chatbot called dr_ant, able to explain the prediction made on the famous Titanic dataset.

Computational argumentation techniques are not limited to the XAI field. For instance, [234] propose an argumentation-based human-agent architecture to map human regulations into a culture for artificial agents with explainable behaviour. Few works in the literature contribute to this subject by providing question banks (e.g. [118, 169]) and design principals (e.g. [59]).

The field of conversational explanations is far from being mature, with the majority of works focusing on conceptual models (e.g. [69]) or single applications, rather than formalising a general model that is applicable to a broader range of explainers. We extend the framework of Madumal et al. [182] to include **clarification dialogues** and propose a system called `ConvXAI`, which can be applied to state-of-the-art explainers and enhance them by providing an interactive dialogical interface.

1.3 Conversational Systems

Keselj [144] defined conversational agents as "programs that communicate with users in NL (text, speech, or both)" and categorised them into task-oriented dialogue agents, which are built to perform a single task (e.g. [76]), and chatbots or open-domains (e.g. [325, 211, 242]), which operate across a wide range of subjects without being dedicated to a specific goal, similar to human–human interactions. Although such a division holds for the majority of works in the literature, several studies have proposed systems that combine these paradigms, e.g. [314]. We review the most common building blocks of conversational systems in the following subsection.

Natural Language Understanding

The general goal of NLU is to understand the general purpose of the query, i.e. intent, and extract the particular slots that the user intends the system to understand from their query concerning the user's intent [139], i.e. entities or slots. While these sub-tasks can be performed separately or jointly, the latter approach provides additional advantages. For instance, one common problem with separate models is to prevent error propagation [103]. Focusing on spoken language understanding (SLU), Chen et al. [57] claimed that joint models "provide rich cues for sentence-level understanding, where both sub-tasks can be mutually improved". Another issue with using separate models for intent classification and slot filling (entity extraction) involves "preserving the hierarchical relationship among words, slots, and intents" [316].

In the next section, we provide an overview of several systems which use neural models, attention-based architectures, or pre-trained language models. Zhang et al. [316] used a

capsule-based neural network (NN) model within a dynamic routing-by-agreement schema, while Guo et al. [107] established a recursive NN by combining a discrete syntactic structure with a continuous-space word and phrase representation to form a compositional model. Xu and Sarikaya [309] used a convolutional NN as a version of the triangular conditional random field [133], while Zhang and Wang [318] employed a gated recurrent unit for this task. Wang et al. [294] designed a bi-model-based recurrent NN semantic frame-parsing structure, which considers cross-impacts through two correlated bidirectional long short-term memory units.

In some of these studies, NNs have been combined with attention layers [279]. For example, Goo et al. [103] established an attention-based recurrent NN model using a slot gate to learn the relationship between intent and slot attention vectors, whereas Liu and Lane [172] developed an attention-based NN. Chen et al. [56] created a joint intent classifier and slot filling model based on the bidirectional encoder representations from transformers (BERT) [73] and conducted an extensive survey of joint models, while Weld et al. [298] performed an in-depth analysis of the current state-of-the-art.

Dialogue State Tracking

In task-oriented conversational systems, dialogue state tracking (DST) is typically employed to determine the state of slots and the user's most recent dialogue act [139]. Note that the "state" in a conversation indicates the "sentiment" of the parties involved in the conversation (e.g. [167]). Žilka et al. [327] identified DST as the core component of any dialogue system that provides a compact representation of the past user input and system output, in the form of a dialogue state, by monitoring dialogue progress. DST ensures that the user has provided all the required information to fill the slots needed to answer their query. For instance, in `ConvXAI`, if a user wants to understand why the model has predicted a particular instance as the current label, they must provide the desired instance index. DST continues to prompt the user for the missing parts until all the previously missing information has been gathered or a certain number of attempts have been made (this maximum number is defined as a system parameter). In this case, the user will be informed that the attempt to retrieve the missing parts has been unsuccessful and encouraged to start over from the beginning.

A common approach for categorising the methods used to create DST systems is to divide them into rule-based, generative, and discriminative models.

Rule-based models Considered the most simple DST method, rule-based models do not require any data for training, which makes them an ideal choice for bootstrapping and provides developers with the opportunity to incorporate domain knowledge [300].

These methods typically rely on hand-crafted rules for tracking the dialogue state, considering the best SLU/NLU result (1-best) as the belief. While rule-based methods reduce the system complexity by limiting the number of states that are being followed, they can mislead other system components, resulting in incorrect outcomes. Although the majority of rule-based methods follow the described strategy in identifying the system state, few have addressed this problem through novel approaches, e.g. [295, 266].

Generative models Generative methods model the dialogue as a Bayesian network, connecting the dialogue state to the system action, user action, and SLU/NLU outcome. Traditionally, generative models enumerated and ranked all the states [300], but increasingly vast state spaces have limited their broader application [308, 113, 128].

Various methods have been proposed for establishing generative DST models. For example, Žilka et al. [327] introduced a generative model that employs a simple dependency structure to achieve fast inference, whereas Zhao et al. [321] studied the impact of pre-training and context representation in designing generative sequence-to-sequence DST models. Serban et al. [253] presented a hierarchical recurrent network generative model and demonstrated its performance against n-gram-based models and vanilla NN models.

Despite the improved performance of generative models over rule-based techniques, they often suffer from one practical shortcoming. Namely, such models often struggle to consolidate large numbers of features from different components like SLU and the conversation history [300]. This leads to the need to explicitly model features, which in turn requires a large amount of data and unrealistic assumptions.

Discriminative models Many authors, including Bohus and Rudnicky [28] as pioneers, have attempted to resolve the shortcomings of generative methods, such as the limitation of modelling correlations between observations in different time slices [162], through a family of techniques called discriminative models. In these models, the class posteriors are directly modelled, allowing a large set of features to be considered, regardless of the

dependencies among them. Unlike generative models, discriminative approaches compute scores for dialogue states using discriminatively trained conditional models [300].

The main difficulty faced by the DST component is caused by the imperfection of automatic speech recognition (ASR) and SLU [300]. As the recipient of the output from the ASR and SLU modules, the DST must be sufficiently robust to identify the true state of the conversation based on the dialogue. This complexity makes the use of traditional methods such as hand-crafted state schema less practical [301].

Dialogue Policy

Given all the required information gathered by DST and considering the contextual information, the dialogue policy (DP) determines which system action (in our case, the explainer and explanation presentation) should be used to answer the user. Commonly, the DP component achieves this by estimating the probabilities using a classifier based on the representations of slot fillers and utterances [139]. Unlike traditional random-sampling methods, modern DP systems mainly rely on reinforcement learning techniques to learn the policies from the user interactions or historical data (e.g. [222, 270, 321, 324]).

Following we list our published on the XAI and conversational XAI subjects:

- Cambria, E., Malandri, L., Mercorio, F., Mezzanzanica, M., & **Nobani, N**. "A survey on XAI and natural language explanations." Information Processing Management 60.1 (2023): 103111.
- Malandri, L., Mercorio, F., Mezzanzanica, M., **Nobani, N.**, & Seveso, A. "ContrXT: Generating contrastive explanations from any text classifier." Information Fusion 81 (2022): 103-115.
- Malandri, L., Mercorio, F., Mezzanzanica, M., **Nobani, N.**, & Seveso, A. "The Good, the Bad, and the Explainer: A Tool for Contrastive Explanations of Text Classifiers." IJCAI. 2022.
- Malandri, L., Mercorio, F., Mezzanzanica, M., **Nobani, N**., & Seveso, A. "Contrastive Explanations of Text Classifiers as a Service." Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: System Demonstrations. 2022.

- **Nobani, N.**, Mercorio, F., & Mezzanzanica, M. "Towards an Explainer-agnostic Conversational XAI." IJCAI. 2021.

1.4 Thesis Structure

The thesis is composed of two main parts, organized as follows:

Part I discusses the role of HITL in XAI and its impact on the end-users perception of explanations. The first chapter provides a brief background on XAI. The next two chapters provide the necessary theory and applications regarding Natural language Explanations: Chapter 2 explores the current state-of-the-art XAI methods and the role of Natural Language explanations while Chapter 3 demonstrates a Natural Language explainer which can be integrated with any black box model. Finally, Chapter 4 introduces a novel system that applies the HITL approach to generate and refine conversational explanations by integrating end-users as a pivotal part of the system.

Part II is devoted to integrating HITL methods with the Labour Market Intelligence field, focusing on taxonomies, using both as input and as the output of discussed AI models. Chapter 6 discusses the topic of embedding evaluation utilizing lexical taxonomies and corpora bonded to such taxonomies, while Chapter 7 shows the application of the HITL approach regarding taxonomy refinement by implementing the evaluation method developed in the previous chapter.

Finally, Chapter 8 contains concluding remarks and discussions about the future directions.

2

Natural Language Explanations

The need for eXplainable AI (XAI) systems is growing as modern Machine Learning (ML) algorithms, particularly "deep learning" ones, are becoming increasingly powerful yet so complex that it is difficult to understand their behaviour and why certain results were achieved, or some mistakes were made. However, understanding the behaviour of those models is as relevant as their performances, allowing users to develop appropriate trust and reliance [121]. The goal of eXplainable AI is to render the behaviour of black box models more understandable, accountable and transparent to humans [43]. This goal can be achieved either by targeting the general decision-making process of a model [50] or by providing insights about a specific outcome [229, 79, 123, 48].

Despite the prominence of XAI methods in recent AI literature and the ever-widening range of their application domains, the attention paid to the *"last mile"* of the XAI-based systems, i.e., the presentation of explanations to end-users is still in a growing phase. This could lead to solutions that, while potent and practical from a technical point of view, cannot be directly utilized by non-expert or non-technical users, defying the principal objective of an XAI system [204]. According to the most recent survey on XAI for machine learning models [43], the communication type of an XAI system can be classified into *textual*,

graphics, and *multimedia* descriptions. While the former uses explanations in a text form and the second is a visual one, the latter combines different types of content like text, graphics, reports, images, audio, video, animation, etc.

Considering the works that the authors are aware of, textual explanations can be expressed by rules [105], codes [192, 148] or natural language explanations [193] and dialogues [132]. The explanations made through natural language, as pointed out by [193], are the key component of future intelligent interactive agents, given their ability to offer interpretability to people with diverse backgrounds and to better mimic humans, which usually explain their decisions verbally [43, 47]. Moreover, in [99], the authors mention that the use of natural language improves decision-making under uncertainty compared to graphical-based presentation methods.

In [260], the authors argue that natural language explanations are suitable for a lay audience given that their interaction mode gives the process a natural feel, while [54] and [70] make a step further by observing that natural language presentation increases the trustworthiness of the explanations and help in garner user acceptance. These explanations are more efficient [11], more insightful for specific cases concerning the visual methods [218] and target a broader range of users. [193] argue that these explanations leverage the common language with the user, profiting from the mental concepts which are already established in the human language [236]. [152] argue that generating a text that mimics how humans use natural language to explain, describe, or inform is not a straightforward task, neither to choose the message communicated nor to transform it to natural language.

It is worth mentioning that the majority of works that use natural language presentation methods (i.e., a small part of the XAI works) utilize primary forms of natural language generation (NLG) techniques [168] like mail-merge (template-filling) which, while effective and easy to control, as [193] point out, some times produce outputs that are non-natural due to their static nature. The minimal attention paid to the presentation techniques by the current state-of-the-art methods and XAI surveys, makes the process of the XAI method selection by researchers and practitioners time-consuming and error-prone, as the XAI literature lacks a consolidated study on the presentation methods and the way an XAI method should be chosen based on its presentation power. For instance, Burkart and Huber [43] allocates a

short paragraph to what they call *communication* or [283] and [284] that only briefly mention textual explanations understandability of the explanations for laypeople.

2.1 Does XAI need Natural Language Explanations?

As was mentioned in the introduction section, the literature report various advantages of using natural language methods in explanation creation.

As mentioned in the introduction section, using natural language methods in explanation creation has various advantages like higher efficiency (See [11]) and coverage (in terms of audience)(See [260]). Another element is the power of these presentation techniques to convey social cues [54], interacting with users and reinforcing their trust in the information system as a whole which goes beyond the black box model and its underneath data [70]. Such potential benefits can be achieved when XAI solutions, starting from static, one-directional messages, go towards dialogues that directly engage the end-user in the explanation process by offering rich and personalized interactions that mimic how humans explain their decisions.

2.1.1 A Roadmap for Selecting XAI-based Systems

The definition of XAI is discipline-dependent [78]. Fields close to social and cognitive sciences tend, when defining explanations, to focus on the problem of providing to the end user sufficient information to establish causation [171, 205] while on the other hand, researchers studying human-computer interactions focus on the interactivity, information transition flow and the effectiveness of explanations [233, 126]. In this chapter, we rely on the definition provided by [238], which relates explanations to ML systems and their components (i.e., independent and dependent variables) and is general regarding the study domain: *"textual or visual artefacts that provide a qualitative understanding of the relationship between the instance's components (e.g., words in a text, patches in an image) and the model's prediction"*.

Paper Selection Criteria. We reviewed 70 XAI papers that make use of natural language. Only those which met these criteria have been included:

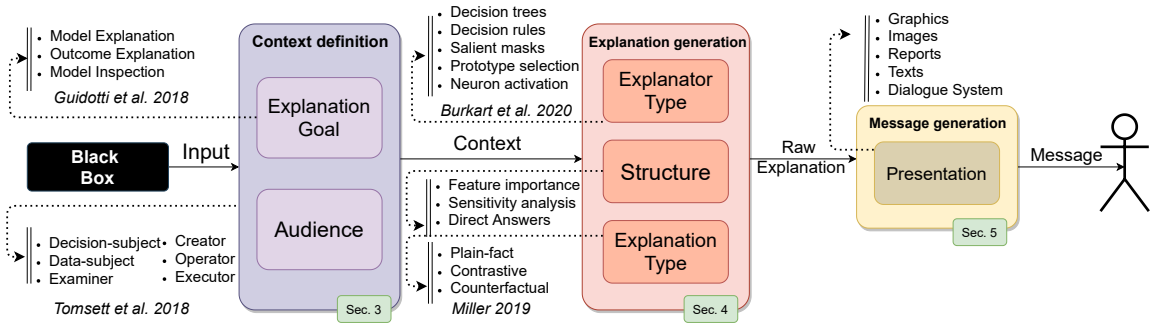


Figure 2.1 A roadmap for selecting XAI systems that make use of natural language explanations

- for journal papers, to be either *Q1* or *Q2* of SCImago journal ranking in any computer science-related topics in the year of publication;
- for conference papers, to be classified as A/B for *all* those rankings: (i) CORE Conference Rating, (ii) LiveSHINE, and (iii) Microsoft Academic.

Google Scholar was searched for papers from 2006 to 2021. We identified search terms as combinations of XAI and NLG set of terms and further extended them, identifying new keywords from those papers. The final term sets are XAI (XAI, Explainable Artificial Intelligence, Interpretable AI, Interpretable Artificial Intelligence, Interpretable Machine Learning, IML) and natural language (Natural Language Generation, Natural Language, NLG, verbalization, text).

Finally, we designed a roadmap for the selection of XAI systems studied based on their characteristics. This roadmap consists of three layers, namely, *Context definition*, *Explanation generation* and *Message generation*, in a way that the input of each layer is the output of the previous one. Sections 2.2 to 2.4 describe these layers in more detail. A similar approach has been proposed in [110], that summarized XAI methods by drawing a list of predefined characteristics. However, our proposed roadmap is different from the mentioned work since (i) it uses categorical measures with simple graphics which are faster both for insertion of new papers and interpretation of method comparison (in contrast to textual descriptions used in the above-mentioned work) (ii) it can be used to compute a rank for each entry based on the relative importance of the features based on the demand of the user. In the following three sections we detail each layer of the roadmap based on the literature review mentioned before.

2.2 Context Definition

In our roadmap, the context indicates a layer of information that identifies how the explainer targets the black box and the need of the end-user (who should use the generated explanations). The context information is used as the input of the explanation generation layer. The building blocks are described below.

2.2.1 Explanation Goal

In their survey, [106] divide XAI problems into two broad categories of black box explanation problems and transparent box design, with the first one further has been categorized into the following three sub-categories:

Model Explanation: Explanations are made through the generation of an interpretable model that tries to mimic the black box, i.e., generating the same output. If such an interpretable model is successful, generating outputs that are similar to those created by the original model, or in other words, have high fidelity to it, such model can be used as a proxy to understand the general decision-making process of the black box. By doing so, we can claim that we globally explained the initial black box.

Outcome explanation: In this case, unlike the model explanations, we are not trying to explain the black box as a whole; instead, given the record, we want to explain its output as a local explanation. Using such a method means we would not comprehend any more the entire mechanism of the black box but only a specific outcome of it.

Model inspection: While the outcome of the previous two methods is a model (an interpretable model) which is able to mimic the black box behaviour (globally in the first and locally in the second case), model inspection, on the other hand, consists of the techniques that instead of generating an interpretable model, provides a visual or textual representation of the model's internal mechanism.

2.2.2 Audience

XAI studies can be divided into two main groups based on how they address the target users. The first group that makes most works are studies that do not mention the target or audience

altogether for their proposed solutions. Almost in all cases, it means that they generate explanations that target technical users who are able to interpret the complex/technical explanations [102, 265, 147]. The second category includes works in which authors mention a general division among different types of audiences (i.e., dividing them into technical and non-technical users), target a specific group of audience or, in rare cases, propose solutions that have a certain level of customization [10, 123].

[271] defines six types of agents - direct and indirect users of an XAI system - in their proposed ecosystem:

- Creator: Agents who create the system, divided into owners and implementors subgroups.
- Operator: Agents that directly interact with the machine.
- Executor: Agents that make decisions based on the output of the AI system.
- Decision subject: Agents that are affected by the decisions.
- Data subject: Agents whose data is used in the targeting of the model.
- Examiners: Agents that audit or investigate the machine.

Similar categories are introduced by previous research, for instance, Bhatt et al., [25] divide what they called *stakeholders* of explainability to Executive, machine learning engineers, end-users and other stakeholders and [157], dividing such stakeholders into five groups of users, (system) developers, affected parties, deployers, and regulators.

Several researchers emphasize the importance of providing explanations that are adequate for the audience, fostering interdisciplinary collaboration to maximize the effectiveness of XAI methods in their context of application [306, 219, 138]. In line with those arguments, we believe that to convey the desired message to the target user successfully and, at the same time, stimulate trust in her, it is necessary to generate explanations tailored to that specific user both in terms of content and form. In our roadmap we use **End users**, **Developers** and **Decision-makers**, as the mostly addressed targets in the literature.

2.3 Explanation Generation

The need for XAI methods is expressed in different forms in the literature; with objectives that sometimes are quite different from each other. Here we mention some works which try to answer the question *What is the necessity of explanations?*:

- Identification of bias improving fairness (e.g., [292, 240, 258, 106])
- Trust in the AI systems and algorithmic decision-making processes (e.g., [122, 258, 178, 1, 106])
- Having better control of the AI systems (e.g., [292, 1])
- Debugging and improvement of black box models (e.g., [143, 292, 258, 207, 154])
- Ethical issues (e.g., [264, 208, 16])
- Legal issues (e.g., [240, 207, 16])
- Improving Transparency (e.g., [299, 16, 1])

Observing from a different angle, [161] identifies the application of explanations in the following fields of AI:

- Machine Learning (except neural networks) (e.g., supervised learning)
- Artificial (Deep) Neural Networks
- Computer Vision
- Constraint Satisfaction and Search (e.g., Conflict resolutions)
- Game Theory (e.g., Zero-sum games)
- Uncertainty in AI (e.g., Probabilistic Graphical Models)
- Robotics (e.g., Information processing)
- Distributed AI (e.g., Multi-Agent Systems)
- Automated Planning and Scheduling (e.g., Unmanned vehicles)
- Natural Language Processing (e.g., Question answering)

Due to its importance, here we briefly describe three types of cognitive processes used in explanations, as outlined by [204]:*Causal connection* or inferring explanation based on the observations and the prior knowledge, *Causal selection* or selecting the inferred explanations and finally, *explanation evaluation* or evaluating the quality of the explanations by the explainee (see e.g., [125]). He further argues that the ideal outcome of the "explanation evaluation" phase, which is the *best evaluation* is not equivalent to choosing the most likely

or the most accurate case, since what is perceived as the best explanation by the explainee is not based on the probability with which the explanation occurs but its pragmatic influence, e.g., usefulness and relevance(see [196]).

2.3.1 Explanator Type

Regardless of the explanation goal one pursues, there are a variety of explanators (i.e., *"part of the AI system which generates explanation artefacts"* [110]). Choosing a model depends on several characteristics of the system like the type of input data and query, accessibility of the black box, its cost and finally, the context of the explanation as described in Section 2.2. [43] did a thorough job identifying the principal explanator types. The most used types are: **Decision Trees** (DT), **Decision Rules** (DR), **Salient Masks** (SM) and **Feature Inspection** (FIN). Here, we briefly describe the most used models in the literature these methods:

- **Decision Tree (DT)**: Being one of the most-used techniques. Decision trees offer both global and local explanations.
- **Decision rules (DR)**: Decision rules describe the inner mechanism of black box models by extracting such rules through various methods. Though it is not technically generated through NLG techniques, decision rules often offer explanations that are easy to understand and interpret by a wide range of users.
- **Salient Mask (SM)**: Mainly used with image data, salient (or saliency) masks cover certain parts of the input to emphasize the segments used for generating the output.

2.3.2 Structure

The general form through which the explainer formulates the explanations can be categorized into the following groups:

- **Feature Importance (FI)**: Providing a complete or limited set of features with their contribution to the final results. As we discussed in Section 2.2, such *"final results"* could refer to a single record's output or the general decision-making logic of the black box model. While used on a large scale, feature importance is not able to demonstrate the root cause of a phenomenon or, in other words, the answer to the *"why"* question.

- **Sensitivity Analysis (SA)**: This kind of explanation can be done for both data features and training parameters. In the first case, the hypothetical output as a result of modifying (adding/removing/altering) the data features will be generated. In contrast, the second case deals with alterations in output as the result of modifications of the black box's internal parameters (e.g., hidden layers in a deep neural network).
- **Direct what/how/why answers (DA)**: This category includes the most intuitive type of explanations, those which directly answer a question of type *what is, what happens if, why rather, how come* and other similar questions (see e.g., [127]).

While being different in forms, we should emphasize that the first two structures mentioned above are special cases, or in other words, limited cases of the latter category. We have divided them into three groups to address the way they are being used in the XAI field. In the majority of works, often the origin of the feature importance and sensitivity analysis remains unmentioned that is, to which question (Why, How, etc.) are they responding, while in fact, many times explanations are rooted in a question which was raised by a specific user with a set of particular needs. We should emphasize that this is not the case for all types of explanations for instance those which are inspired by mathematical equations (See e.g. Layer-wise Relevance Propagation (LRP) [19]).

In our opinion, not elaborating on the choice of the explanation structure can damage the overall effectiveness of the XAI system, as the wrong structure will risk the knowledge transferring process as the final goal of any XAI system.

2.3.3 Explanation Type

Working on explanations applied in information systems (IS), [129] proposes an expanded concept of explanations, arguing that the choice of explanation types depends on the reference disciplines through which research phenomena are understood and the research agenda is shaped. Below we briefly introduce these explanation types by providing a general form for each of them:

- **Covering-law(deductive-nomological) explanations**: "Whenever phenomenon X is observed to occur in the setting of conditions C, Y will be observed."
- **Statistical-relevance explanations**: "Based on empirical data, factors A, B and C contribute to the probability of Y by the amount of X."

- **Contrast-class explanation:** "In this context and given my purpose, why did X (rather than X*, X**, etc.) occur?"
- **Functional explanations**: "Identification of the mechanism by which desirable goal A ensures the continued existence of the phenomenon."

As the result of his survey, [204] argues that a vast amount of such works (e.g., [72, 141]) are based on the four "modes of explanation" proposed by Aristotle, which are: Material (a substance which makes something), Formal (Form of something which its identity depends on), Efficient(the proximal mechanism cause a change) and Final (the end goal of something). Miller further declares that *"explanations are contrastive"* and throughout his work, confronts it with *"complete explanations"* which, unlike the former, respond to straightforward **plain-fact** questions of the type *"why does object a have property P?"*, by listing the entire causal chain which results in the observed output.

Contrastive explanations (e.g., [171, 205, 204]) are the natural response to a *why* questions, while some argue that *how* and *what* questions are also considered as such. [204] points out that contrastive explanations provide a window in the questioner's mental model by showing their knowledge gap while at the same time, these explanations, with respect to the complete explanations discussed earlier, are more straightforward, more feasible and cognitively less demanding for both parties engaged in the explanation process. Different types of contrastive explanations are introduced in the literature and we go through some of them in the following part, but before doing so, we should mention what parts these different proposals of contrastive explanation have in common: Contrast class, fact and foil. For these concepts, we rely on the definition done by [241]: *"A contrast class F are all possible alternatives to a decision given the context (i.e., the range of values for a decision E). The fact is the actual decision $f \in F$, while the foil is any other member of the contrast class that is not fâ, i.e., $g \in F \setminus \{f\}$".* As noted by [197], such definition of the counterfactuals, as the hypothetical outcome for event E, hold only for contrastive explanations while the same concept in the causality and its closely related concept, causation, is a "non-cause" in which the event-to-be-explained [204] does not occur. [276] goes further and introduces three types of contrastive questions:

- **P-contrast**: Why does object a have property P, rather than property Q?
- **O-contrast**: Why does object a have property P, while object b has property Q?

- **T-contrast**: Why does object a have property P at time t, but property Q at time t?

As [205] puts it, P-contrast - or the standard *"rather than"* question - happens within an object, O-contrast among objects themselves and T-contrast within an object over time. Furthermore, using the framework of [112], Miller categorizes the concept of P-contrast as "Alternative explanations" while labels O-contrast and T-contrast concepts as "Congruent questions", formalizing them in the [205].

[311] provides another classification for contrastive questions as "incompatible" and "compatible" cases, while the former is when fact and foil are inconsistent and unlike the fact, the foil does not happen and is hypothetical (similar to P-contrast mentioned above), while in the latter case, fact and foil (or as he calls it, surrogate) are compatible and they both happen in diverse situations/times.

Finally, **Counterfactual** explanations answer to questions about the hypothetical outcome of a hypothetical event, or as [286] puts it, *"how the world would have to be different for a desirable outcome to occur"* (see e.g., [44, 282]).

2.4 Message Generation

Given the focus of this chapter on the usage of natural language techniques in the XAI field, we provide a more detailed description of NLG techniques and dialogue systems as a presentation category and a sub-category of the text presentation.

The Explanation generator layer provides explanations of the black box that cannot be delivered directly to the end user as they are in a raw format, often using model-dependent notations. Hence, the role of this layer is to transform these raw outputs to explanations (e.g., *Messages*) that are comprehensible by the end-user.

2.4.1 Presentation Technique

One of the less explored aspects of XAI is the presentation layer, where the explanations made by the explainer are transmitted to the end user. The output of an XAI system can be multimodal, thus presenting natural language explanations and other content types. We grouped the presentation methods together: **Graphics/plots**, **Texts**, **Images** and **Reports**.

The choice of presentation methods depends on various interrelated factors. In our opinion, the most contributing ones include the ease of producing the representations (e.g., out-of-the-box solutions) and overlooking the importance of the presentation method on users' comprehension [131]. In the following part, we briefly describe the typical presentation methods in the literature.

Graphics/Plots contain the most popular methods in the literature. Such popularity in our opinion is rooted in the presence of tools and the relative simplicity of generating such graphics. This group mainly consists of the following types: *Bar plots*, *Line plots*, *Trees*, *heatmap plots*, *histograms*, *scatter plots* and *bubble plots*.

Bar Plot is the most used method in this category and can be further divided into Horizontal and Vertical Plots [238, 226, 177]. Figure 2.2 provides an example of a bar plot.

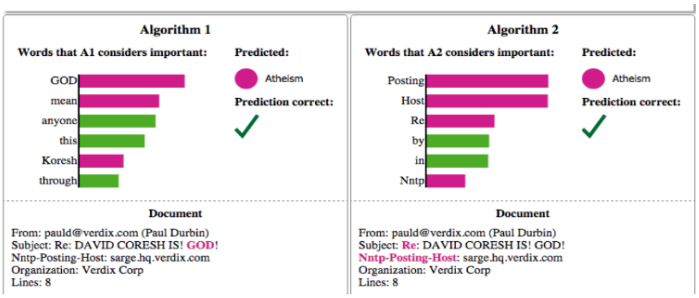


Figure 2.2 An example of a bar plot from [238]

Line Plot vary from simple vertical bar plots to sophisticated custom plots made to represent a particular subject, often mixed with other types of methods like destiny plots or changing hue for adding additional attributes [3, 213, 102]. Figure 2.3 provides an example of a line plot.

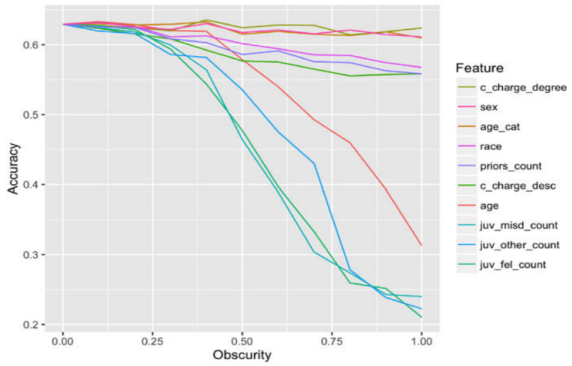


Figure 2.3 An example of a line plot from [3]

Trees can be divided into two main categories: (i) Boolean Rules Trees which use the logic decision gates to classify records, and (ii) Decision Trees which utilize the Boolean decisions instead [137, 194, 142]. Figure 2.4 provides an example of a tree.

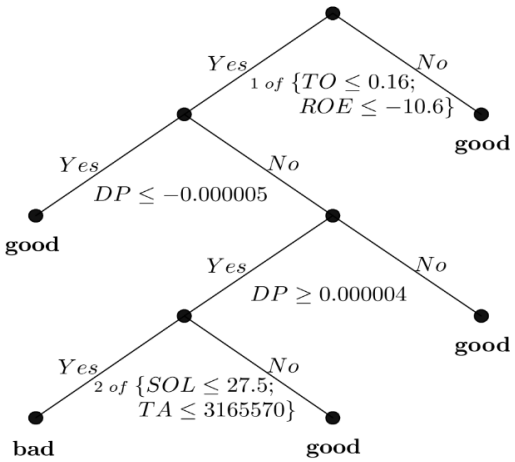


Figure 2.4 An example of a tree from [194]

Heatmap Plot (not to be confused with Heatmap Images) are rather simple visual presentations that map a numeric value to its corresponding color [315, 252]. Figure 2.5 provides an example of a heatmap.

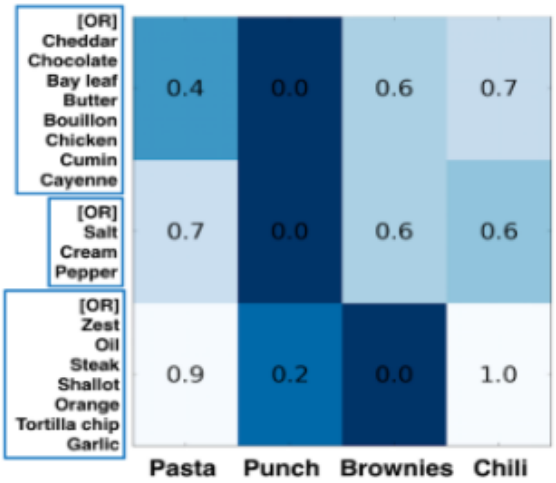


Figure 2.5 An example of a heatmap from [147]

Histogram among the presented methods so far are the most technical methods as their interpretation might be complicated for the layman user without statistical knowledge [20, 213]. Figure 2.6 provides an example of a histogram.

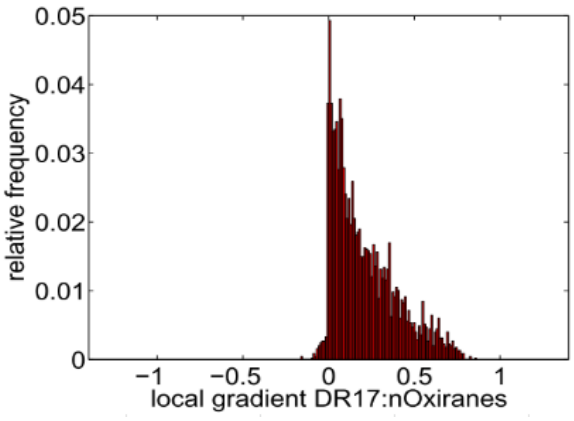


Figure 2.6 An example of a histogram from [20]

Scatter Plot , often boosted with other visualization methods, is used to map two or more dimensions into two or three-dimension space [20]. Figure 2.7 provides an example of a scatter plot.

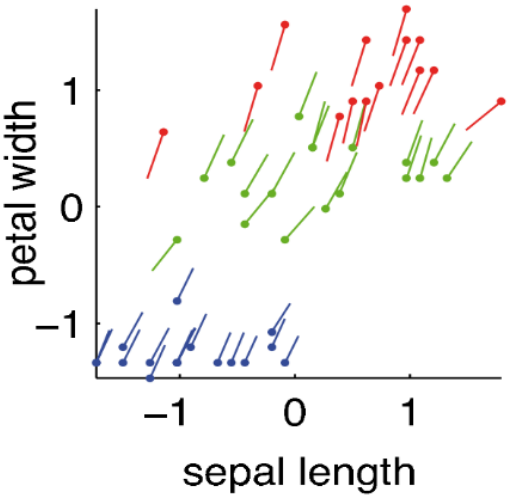


Figure 2.7 An example of a histogram from [20]

Bubble Plot, visually similar to scatter plot, can be considered as an augmented version of scatter plot and is often used to combine categorical and continuous values [275]. Figure 2.8 provides an example of a bubble plot.

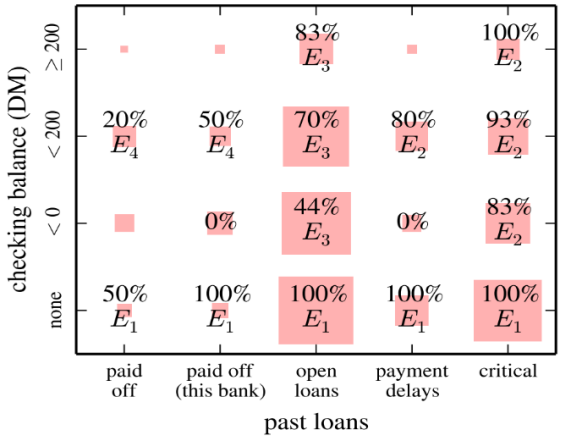


Figure 2.8 An example of a bubble plot from [282]

Images

Image-based presentations are considered more sophisticated with respect to the previous group (plots/graphics) and, at the same time, are more limited since they can be applied only if the target input is an image. The main types of this category are *image heatmap*, *saliency masking*, and *image manipulation*.

Image Heatmap, not to be confused with heat map plots, use an image as their base and add different layers of visualization, mostly coming from continuous data [147]. Figure 2.9 provides an example of a heatmap.

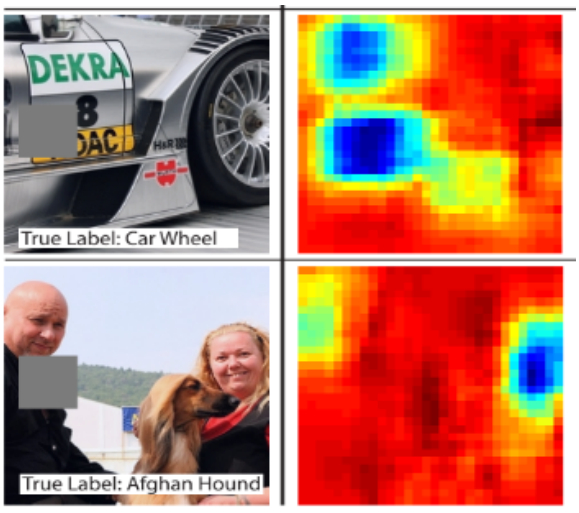


Figure 2.9 An example of a heatmap from [315]

Saliency Masking is similar to image heatmaps as they utilize an image as their basis, but instead of adding heatmaps of values, they partially mask/cover the image to communicate a specific message [238, 19, 256]. Figure 2.10 provides an example of a saliency masking.

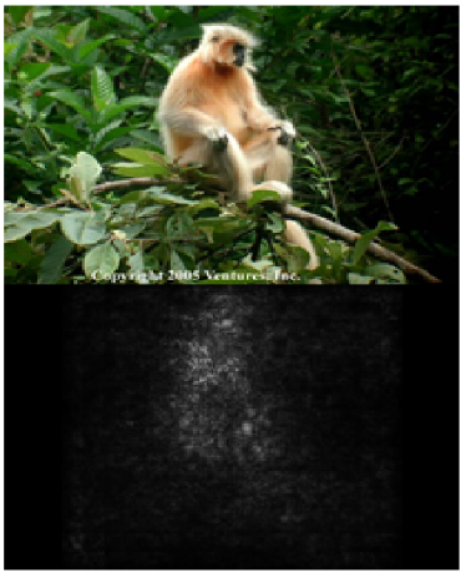


Figure 2.10 An example of a saliency masking from [255]

Image Manipulation, being the less sophisticated method in its family, image-manipulation consists of adding indicator shapes to an image in order to indicate a specific part of the image [275, 239]. Figure 2.11 provides an example of an image manipulation.

Figure 2.11 An example of an image manipulation from [282]

Reports

Although this family is close the *text* category, described below, reports have a more structured approach respect to texts and often are combined with other methods (e.g., graphics). The main techniques in this category are: *tabular reports*, *decision tables* and *graphical table reports*.

Tabular Report. The most basic method of this family, reports, conveys the desired message in a structured and direct manner [226, 117]. Figure 2.12 provides an example of a tabular report.

Color	Feature	Value	Contribution
	plasma membrane (SE) (RE)	1	3.14499
	transmembrane (SE) (RE)	1	2.69372
	inner membrane (SE) (RE)	-1	-1.77431
	prior (SE) (RE)	constant	-1.76159
	cytoplasmic (SE) (RE)	-1	1.33019
	integral membrane protein (SE) (RE)	-1	-1.18103
	plasmid (SE) (RE)	1	0.80206
	nitrogen fixation (SE) (RE)	1	0.70888
	protein biosynthesis (SE) (RE)	-1	0.62916
	+ Aggregate (130)	+	2.86909
	- Aggregate (265)	-	-3.47801
	Decision	+	3.98314

Figure 2.12 An example of a tabular report from [268]

Decision Table. Like tabular reports, decision tables use the tabular structure, but since they solely represent the rules and mostly, no other info, they have less flexibility in the data types and other representations [281]. Figure 2.13 provides an example of a decision table.

1. Intl_Plan	0														1			
2. Intl_Calls	≤2	>2													-			
3. Intl_Mins	-	≤13.15				>13.15									-			
4. CustServ_Calls	-	-				≤3								>3	-			
5. Vmail_Message	-	≤2			>2	≤2						>2		-	≤2			>2
6. Eve_Mins	-	-			-	≤248.15			>248.15			-		-	-			-
7. Day_Mins	-	≤285.5	>285.5		-	≤285.5		>285.5		-		-		-	≤285.5	>285.5		-
8. Day_Charge	-	-	≤48.53	>48.53	-	-		≤48.53		>48.53	-	-		-	-	≤48.53	>48.53	-
9. Intl_Charge	-	-	-	-	-	≤3.55	>3.55	≤3.55	>3.55	-	-	≤3.55	>3.55	-	-	-	-	-
1. class=churn	x	-	-	x	-	-	x	-	x	x	x	-	x	x	-	-	x	-
2. class=non churn	-	x	x	-	x	x	-	x	-	-	-	x	-	-	x	x	-	x
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Figure 2.13 An example of a decision table from [281]

Graphical Table Report. This method, using tabular reports as the basis, integrates other methods in a very flexible way which allows one to customize the table based on the specific message desired to be communicated [146]). Figure 2.14 provides an example of a graphical table report.

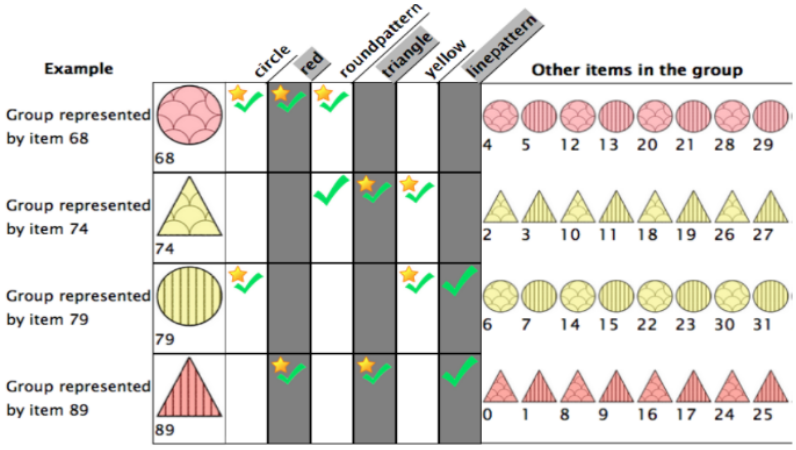


Figure 2.14 An example of a graphical table report from [146]

Texts. This group contains methods that use the text as their basis. Notice that it does not mean that the output of these representations are necessarily expressed in natural language but indicates that the main message is conveyed through the text and not the other techniques mentioned before. The main textual representations are *rules*, *word annotations*, and *natural language* texts. Figure 2.15 provides an example of a textual explanation.

a beer that is not sold in my neck of the woods , but managed to get while on a roadtrip . poured into an imperial pint glass with a generous head that sustained life throughout . nothing out of the ordinary here , but a good brew still . body was kind of heavy , but not thick . the hop smell was excellent and enticing . very drinkable

very dark beer . pours a nice finger and a half of creamy foam and stays throughout the beer . smells of coffee and roasted malt . has a major coffee-like taste with hints of chocolate . if you like black coffee , you will love this porter . creamy smooth mouthfeel and definitely gets smoother on the palate once it warms . it 's an ok porter but i feel there are much better one 's out there .

i really did not like this . it just seemed extremely watery . i dont ' think this had any carbonation whatsoever . maybe it was flat , who knows ? but even if i got a bad brew i do n't see how this would possibly be something i 'd get time and time again . i could taste the hops towards the middle , but the beer got pretty nasty towards the bottom . i would never drink this again , unless it was free . i 'm kind of upset i bought this .

a : poured a nice dark brown with a tan colored head about half an inch thick , nice red/garnet accents when held to the light . little clumps of lacing all around the glass , not too shabby . not terribly impressive though s : smells like a more guinness-y guinness really , there are some roasted malts there , signature guinness smells , less burnt though , a little bit of chocolate ... ... m : relatively thick , it is n't an export stout or imperial stout , but still is pretty hefty in the mouth , very smooth , not much carbonation . not too shabby d : not quite as drinkable as the draught , but still not too bad . i could easily see drinking a few of these .

Figure 2.15 An example of a decision table from [163]

Another subcategory of AI which can mitigate the lack of explicit, symbolic representation of knowledge, i.e what prevents humans from fully comprehending black boxes is Symbolic AI ([60]). In this subcategory of AI the output can be in the textual/code (See [192, 148]). Inductive Logic Programming (ILP), a subfield of symbolic AI and more specifically a technique called Learning from Interpretation Transition (LFIT) can learn a propositional logic theory equivalent to a given black box system under certain conditions [214]. Another example of Symbolic AI can be seen in the work of [188] that uses Binary Decision Diagram (BDD) to derive T-contrast explanations for text classifiers.

Natural Language Explanations. As part of text explanations, natural language explanations are text written in plain English or other human languages(see e.g., [114, 115]). Most of the approaches for generating explanations in natural language belong to the families of *NLG* and *Dialogue Systems*. Notice that, while the sentence generation task in dialogue systems is an application of NLG, they are more closely related to dialogue management since management and realization policies are usually learned together [89]. For this reason, we treat them as two different types of output.

NLG. In the seminal work of [235], NLG is defined as *"The sub-field of artificial intelligence and computational linguistics that is concerned with the construction of computer systems that can produce meaningful texts in English or other human languages from some underlying non-linguistic representation of information"*. In essence, NLG is a branch of natural language processing (NLP) research [206, 322, 49] focusing on the transformation of computer language into natural language. Traditionally, NLG tasks are divided into two main categories; data-to-text and text-to-text. As the name suggests, the data-to-text group deals with generating natural language mainly from numerical data. As examples of this category, we can mention Robo-Journalism or automatic reporting (e.g., automatic weather forecast [263] and sports event reports [55]). Text-to-text category, on the other hand, covers a wider and somehow more significant applications like machines translation (e.g., [149, 74]), text summarization and simplification [296] and paraphrasing [14]. While the mentioned categories of NLG are the major players in the NLG field, in the past decade, another group, vision-to-text, has emerged, mainly thanks to the proliferation of the Deep Neural Network methods. Although this category is not yet as mature as the methods mentioned above,

there already exist numerous applications like image captioning [310] and the generation of natural language explanations using Deep Learning techniques (e.g. [67, 79, 53]) As [193] clarifies, NLG works can also be divided considering the technology used for generations of the text: *Template-based*, which structure templates that present the output in textual form and *End-to-end* generation which utilizes large humanly labelled data-to-text corpora.

Table 2.1 Mapping selected papers to our roadmap. (Example and Benchmark) → Not provided/used: □, Provided/used once: ⧠, Provided/used multiple times: ■ ; (Dataset) → Not mentioned: 👁‍🗨, Private dataset: 🔒, Public dataset: 🔓; (Code) → Not provided: 🗝, Provided no documentation: git, Provided with documentation: git; (Rest of features) → Not mentioned: ○, Mentioned but not applied: ◉, Applied: ●

	Experiments				Evaluation		Explanation Goal			Audience			Explanator Type				Structure			Explanation Type			Presentation				
Paper	Example	Benchmark	Dataset	Code	User-Evaluation	Metric	Model Explanation	Outcome Explanation	Model Inspection	end user	Developer	Decision-Maker	Decision Tree	Decision Rule	Salient Masks	Feature Inspection	Feature Importance	Sensitivity Analysis	Direct Answers	Plain-fact	Contrastive	Counterfactual	Graphic	Image	Rule	Text	Dialogue System
Costa et al. (2018)	■	□	🔓	🗝	○	●	○	○	●	●	○	○	○	○	○	◉	○	○	●	●	○	○	○	○	○	●	○
Ehsan et al. (2019)	■	□	👁‍🗨	🗝	●	●	○	●	○	●	○	○	○	○	○	○	○	○	●	●	○	○	●	○	○	●	○
Chang et al. (2016)	■	⧠	🔓	🗝	●	○	○	○	●	●	○	○	○	○	○	○	●	○	●	●	○	○	○	○	○	●	○
Hendricks et al. (2018) a	■	□	🔓	🗝	○	●	○	●	○	○	○	○	○	○	●	○	○	○	●	○	○	●	○	●	○	●	○
Alonso et al. (2019)	■	□	🔓	git	◉	○	●	●	○	●	●	◉	●	○	○	○	○	○	○	●	○	○	●	○	○	●	○
Sokol et al. (2018)	⧠	□	🔓	🗝	○	○	○	●	○	○	○	○	●	○	○	○	○	◉	●	●	●	●	○	○	○	○	●
Core et al. (2006)	■	□	👁‍🗨	🗝	○	○	○	●	○	●	○	○	○	◉	○	○	○	○	●	●	○	○	○	○	○	○	●
Rosenthal et al. (2016)	■	□	👁‍🗨	🗝	○	●	○	●	○	●	●	○	○	○	○	○	○	○	●	●	○	○	○	○	○	●	○
Amarasinghe et al. (2019)	■	□	🔓	🗝	○	●	●	○	○	●	○	○	○	◉	○	○	○	◉	○	●	○	○	○	○	○	●	○
Hohman et al. (2019)	■	□	👁‍🗨	🗝	○	○	●	●	●	●	●	●	○	○	○	○	●	○	○	●	○	●	●	○	○	●	○
Park et al. (2018)	■	⧠	🔓	🗝	●	●	○	●	○	●	○	○	○	○	●	○	○	○	●	●	○	○	○	●	○	●	○
Malandri et al. (2022)	■	⧠	🔓	🗝	○	●	●	○	○	●	○	◉	●	●	○	○	○	○	●	○	●	○	●	○	●	●	○
Zhao et al. (2021)	■	⧠	🔓	git	○	●	○	●	○	●	○	○	○	○	●	●	●	○	○	●	○	○	●	●	○	○	○
Donadello et al. (2021)	■	□	🔓	git	○	○	○	●	○	●	○	○	●	○	○	●	●	○	●	●	○	○	○	○	○	●	○
Hendricks et al. (2016)	■	⧠	🔓	🗝	●	●	○	●	○	●	○	○	○	○	○	●	●	○	○	●	○	○	○	○	○	●	○
Hendricks et al. (2018) b	■	⧠	🔓	🗝	●	●	○	●	○	●	○	○	○	○	○	●	●	○	○	●	○	●	○	●	○	●	○
Sreedharan et al. (2021)	■	⧠	🔓	🗝	●	●	○	●	○	●	○	○	○	○	○	●	○	○	●	○	●	○	○	○	○	●	○

Rules. Rules are simply a list of decision rules written in natural language [326, 194, 164].

Word annotation. Word annotation is the fastest and simplest method of the text category, in which a message is conveyed by highlighting or changing the colour of a specific part of the text [173, 238, 163, 146].

Dialogue Systems. Essentially, a dialogue system is a system that enables the conversation between two parties. Neither the term *dialogue system* nor its definition, however, have a clear consensus among the researchers. While there are various alternatives for this term, we can mention *conversational Agent*, *Conversational User Interface* and *Chatbots* are the most commonly used in academia and business [307, 313, 312, 180, 167]. Although such systems have been around for the past fifty years, their usage as a presentation layer in XAI systems is minimal. [120] defines an explanation as *"someone explains something to someone else"*, which emphasizes the conversation form of a causal explanation. What we saw in the proposed roadmap until this point was generating explanations using context and presenting them as a fixed explanation. An alternative to such presentations is to communicate the message (explanation) as a part of a conversation between the system (explainer) and the user (explainee).

2.5 Keeping the roadmap up-to-date

One of the significant limitations of survey studies is that they rapidly become obsolete as soon as the research on the topic advances. Aiming at overcoming this issue, [230], propose an interactive browser-based system called XNLP[1] which *synthesizes the state of the field at different levels of abstractions and from different perspectives*. Although this tool, in our opinion, brings value to the XAI community by providing a dynamic hub of recent works, it lacks a feature that directs researchers to the most related works to their field of research.

To bridle this limitation, we propose to model the roadmap depicted in Figure 2.1 as a multi-criteria-decision-making (MCDM) problem where columns of Table 2.1 are criteria whilst the rows are the alternatives on which decide. Hence, the decision goal is *to decide which is the most suitable XAI system that makes use of natural language explanations*.

[1] https://xainlp2020.github.io/xainlp/home

Modelling such a decision as an MCDM problem allows deciding, taking into account the user needs (criteria) and their relative importance of them (weight of criteria).

2.5.1 Multi-Criteria Decision-Making at a glance

In essence, MCDM refers to a set of methods that allows constructing a global preference relation for a set of alternatives to be evaluated by using several criteria. A literature review on MCDM falls out of the scope of this chapter; the reader can refer to [84] for a survey. The MDCM approaches are able to deal with dependence amongst criteria (e.g., ANP [246]), conflicting criteria (ELECTRE), synthesize compromise solutions (TOPSIS), as well as to deal with uncertainty over the judgments (Fuzzy sets theory applied to the previous methods). In our work, we use the *Analytic Hierarchy Process* (AHP) [245], as it is beneficial for evaluating complex multi-attribute alternatives involving subjective criteria to capture stakeholders' knowledge of phenomena under study. AHP consists of the following main steps.

(i) Build up the criteria/alternatives tree. In this step, the criteria that compose the decision problem are identified and organized hierarchically so that a criterion may have sub-criteria, and so on. The leaves of this tree are the alternatives that the decision process aims at selecting. Our hierarchy of criteria is drawn following Table 2.1.

(ii) Pairwise Comparison of Criteria. In this step, the users are required to perform a pairwise comparison of each criterion at each level of the hierarchy, and the results are collected in a matrix summarizing the local priorities for each domain expert. The main intuition here is that it is easier (and more accurate) to compare the importance of two criteria at a time than simultaneously evaluating all of them. There are two characteristics of AHP that deserve to be highlighted. First, the same preference scale, i.e., the Saaty's Scale [246], is used to evaluate both (quantitative and qualitative) criteria and alternatives. Second, the expert does not provide any absolute numerical judgment but a comparative evaluation, which is more familiar to people. Comparisons are recorded in a positive reciprocal matrix, in which a_{ij} represents the comparison between element i and j.

The rationale of the relationship $a_{ji} = 1/a_{ij}$ is that if A is four times more important than B, then B is $1/4$ important with respect to A. Thus, if the matrix is perfectly consistent,

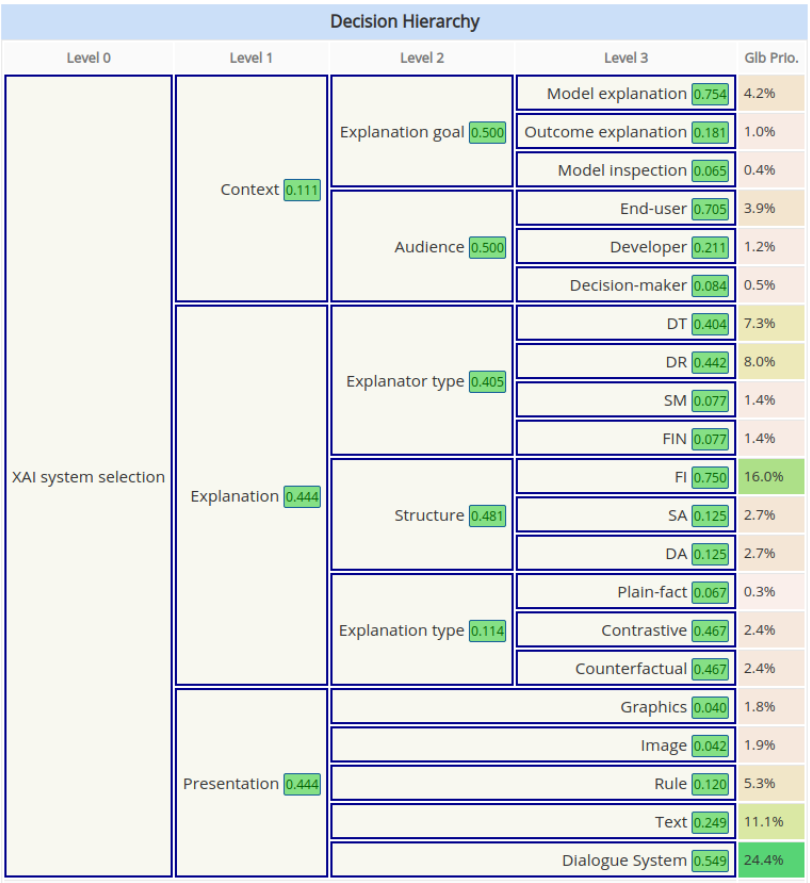


Figure 2.16 The AHP hierarchy built from Tab.2.1 weighted by a user. Any user can contribute weighting the hierarchy at https://tinyurl.com/XAI-NLG-AHP

the transitivity rule is satisfied for all the comparisons, namely $a_{ij} = a_{ik} \cdot a_{kj}$. Intuitively, it is expected that if A is *moderate important* (3) than B, and *B* is *weak important* (2) than C, thus a consistent judgment would have that A is $3 \cdot 2 = 6$ *strong important* than C. As inconsistencies are natural in human judgments, AHP provides the consistency ratio to the final user. It was proved that inconsistencies in answers could be tolerated if the consistency ratio remains within a small interval, that is 10% [246].

At the end of this process, a weighted hierarchy that encodes the user preference is obtained, as in Figure 2.16. Notice that AHP allows group decision-making by averaging judgments into one unified weighted hierarchy.

(iii) Synthesize Global Priorities of Alternatives. The last step requires synthesizing the global priorities (i.e., the priority vector) from the pairwise comparisons to determine the ranking of *alternatives*, taking into account the user judgments computed in the previous

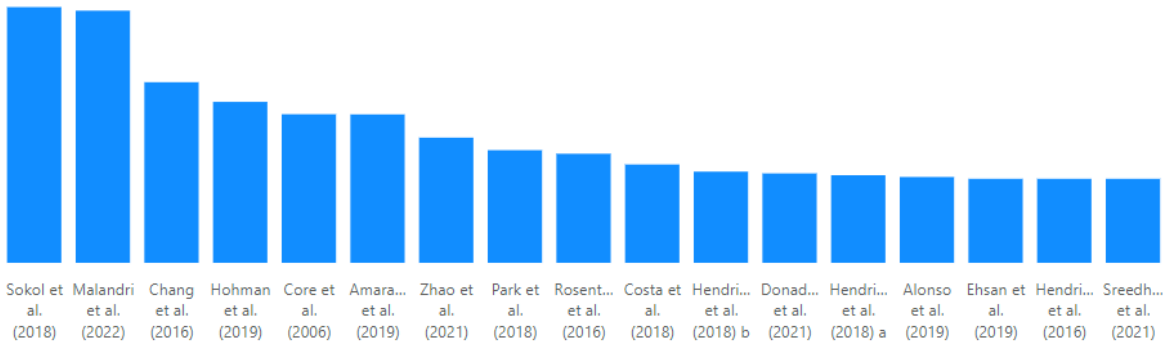


Figure 2.17 The paper ranking based on the hierarchy shown in Figure 2.1

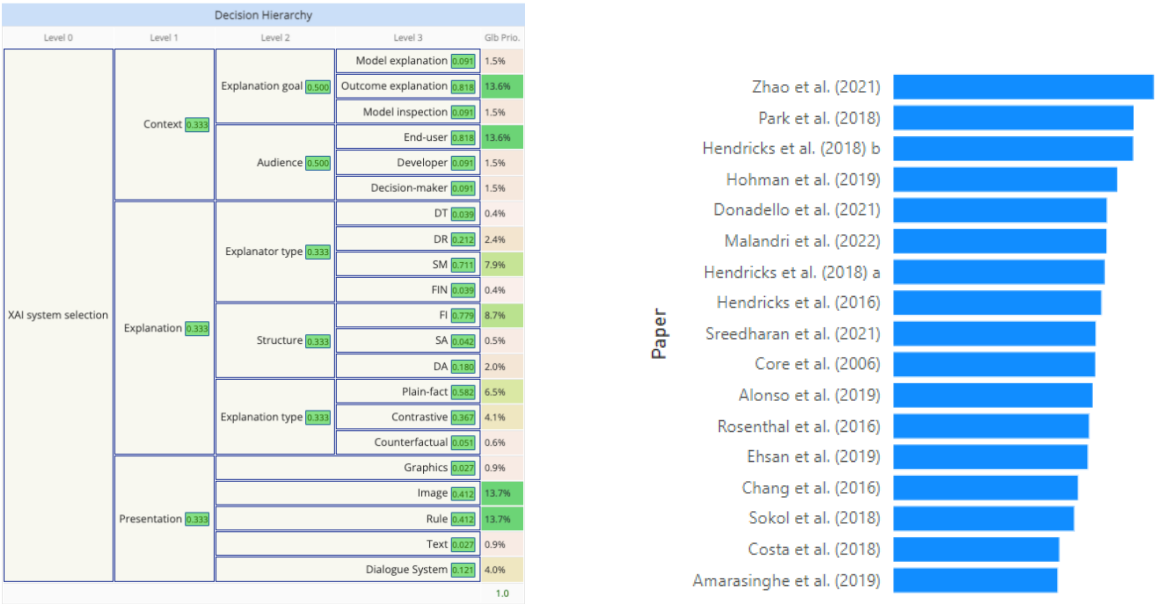
step. Mathematically speaking, the priority vector is the solution of an Eigenvalue problem over the matrix previously introduced. The results of the pairwise comparisons are arranged in a matrix. The matrix's first (dominant) normalized right Eigenvector gives the ratio scale (weighting), while the Eigenvalue determines the consistency ratio. A the end of this step, a list of alternatives ranked is provided, as in Figure 2.17; the figure shows the paper rankings user got based on the created hierarchy, after a pairwise comparison of papers. In our approach, once the weighted hierarchy of criteria is obtained, the pairwise evaluation of *alternatives* is automatically performed drawing from Table 2.1, by normalizing the values on the Saaty's scale. One might note that the user assigned 44% of importance to both the Presentation and Explanation layers, whilst the Context account for the 11%. Looking at global priorities, having a *Dialogue System* accounts for the 24.4% on the final decision, as well as being able to provide Feature Importance (FI) accounts for the 16% globally, more than having *text* (11.1%).

Based on these results, the paper that better fits the preferences of the decision hierarchy in Figure 2.17 is [260] with the consolidated weight of 14.3%. This paper can explain this output is one of two using *dialogue systems*, which have the highest priority in the hierarchy defined by the user. The reason why the output is not [65] - the other paper using a dialogue system - is that, unlike the latter, [260] uses Decision Trees which is another relevant criterion in the hierarchy. In essence, AHP allows one to capture and keep track of the reason behind the decision, taking into account the relative importance of the XAI characteristics.

One should note that users are able to update the roadmap criteria and add new alternatives (papers) to adapt the framework and update it based on their specific needs. In order to demonstrate how the framework transforms specific users' needs into paper rankings, in the

following part we provide three simulated cases including the initial need, hierarchy weights of criteria and paper rankings.

Case 1 Working with a dataset which has both images and separate features the researcher's goal is to classify EMG hand movement. To do so, the researcher wants to be able to explain individual outcomes of the black box to final users. Such explanations should be able to satisfy both plain-fact (why questions) and contrastive (why not or why this and not that) questions of end-users, using either rules or images. It's also acceptable to receive such explanations through a conversation.

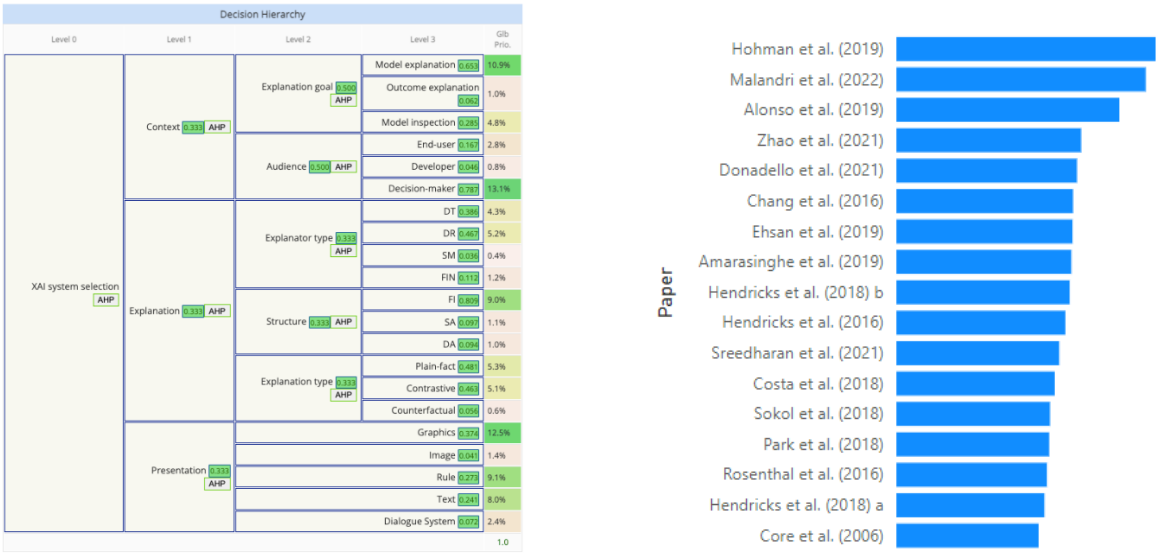


(a) A (b) B

Figure 2.18 Hierarchy (A) and paper ranking (B) of case 1

Case 2 The researcher's goal is to classify job vacancies and have a clear understanding of the characteristics of the classification. The explanations are destinated for the decision-makers who act upon the obtained results. The preferred mode for explanations to be conveyed are decision trees and rules while the preferred presentations are through rules, graphics or natural language.

Case 3 Having tabular data of credit landing the researcher aims to inspect the model and assess its fairness. The target of the explanations are the developers and the most efficient

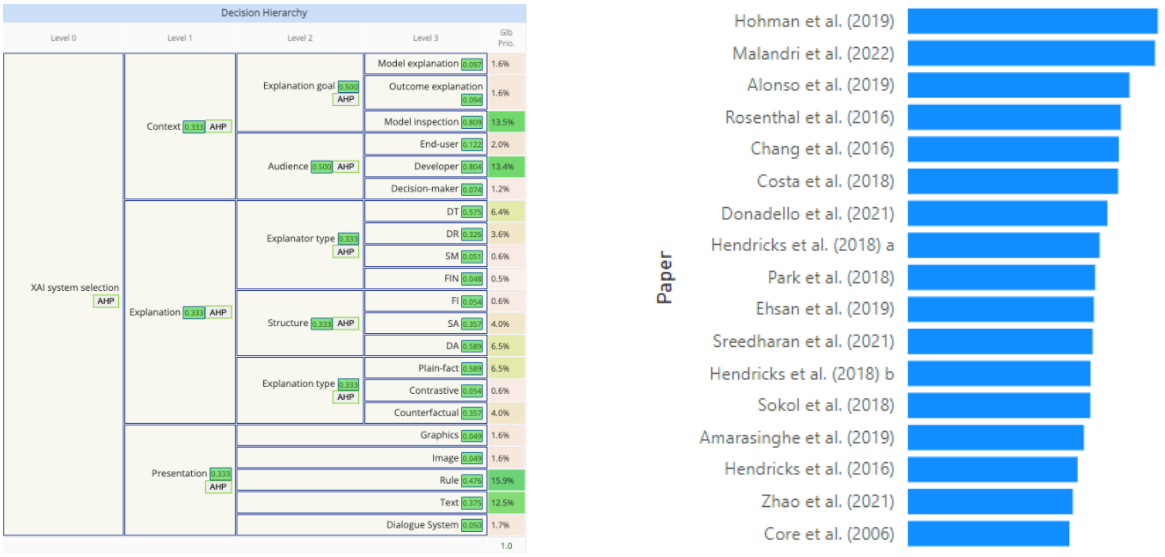


(a) A (b) B

Figure 2.19 Hierarchy (A) and paper ranking (B) of case 2

way for them to comprehend and act upon them is through decision rules and trees and in form of plain-fact and counterfactual explanations. Similarly, the preferred modality of presentation for this specific target is textual or rule explanations.

Discussion Figures 2.18, 2.19 and 2.20 show the decision hierarchy and the resulting ranking of papers based on these preferences. As can be observed, the paper ranking accurately reflects the preferences of researchers in terms of explanation goals, audiences and presentation choices. For instance, in case 1, the researcher working with image data prefers to receive the explanations in a plain-fact manner and thorough images, which is aligned with the paper ranking provided by the tool, with [323] as the first paper. Similarly, in case 2, where the preferred explanations are rules and natural language, the top-ranked papers are [123] and [188] which provide such explanations. Finally, in case 3 the user is a developer that needs an XAI algorithm able to process tabular data and provide explanations in the form of rules and trees, like the proposed ones. Note that, given that both case 2 and case 3 require decision rules and trees as explanations, the first three suggestions point to the same methods, while the following differs. For instance, the verbalization model proposed by [243] is not designed for text classification. Therefore, it appears as the fourth best fit for case 3, but it is much lower in the ranking for case 2. Those examples show the usefulness of



(a) A (b) B

Figure 2.20 Hierarchy (A) and paper ranking (B) of case 3

the proposed approach, which helps the users in narrowing the search among the vast range of available XAI methods, pointing to the one that better fits their needs and tasks.

3

Explaining black box Classifiers Through Natural Language

Continuing the argument we started in Chapter 2, that is the role and the importance of NL explanations, in this chapter we demonstrate `ContrXT`, a novel tool that computes the differences in the classification logic of two distinct trained models, reasoning on their symbolic representation through Binary Decision Diagrams and uses NL to present the explanations.

Motivating Example

The example below is inspired by a real-life problem in the field of text classification of multilingual online job ads within an EU project [52, 80]. To clarify the matter, let us consider an organisation that needs to classify millions of online job ads to analyse labour market dynamics over time across borders. In such a scenario, training an ML model would be helpful to support questions such as: *Which occupations will grow in the future and where? What skills will be demanded the most in the next years?* However, once such an ML model has been trained and deployed (see, e.g., [64, 32]) it needs to be periodically re-trained as the labour market demand constantly changes through time, mainly due to rise of new emerging

occupations and skills [94, 95]. This, in turn, leads policy makers to ask if - and to what extent - the re-trained model is coherent in classifying new job ads with respect to the past criteria.

3.1 ContrXT in a Nutshell

Consider a text classifier ψ_1, retrained with new data and resulting into ψ_2. The underlying learning function of the newly trained model might lead to outcomes considered contradictory by the end users when compared with the previous ones, as the system does not motivate why the logic is changed. Hence, such a user might wonder *"why the criteria used by* ψ_1 *result in class c, but* ψ_2 *does not classify on c anymore?"*.

This is posed as a *T-contrast* question, namely, *"Why does object A have property P at time* t_i*, but property Q at time* t_j*?"* [204, 276].

ContrXT (Contrastive eXplainer for Text classifier), computes model-agnostic global T-contrast explanations from any black box text classifiers. ContrXT, as a novelty, (i) encodes the differences in the classification criteria over multiple training phases through symbolic reasoning, and (ii) estimates to what extent the retrained model is congruent with the past. ContrXT is available as an off-the-shelf Python tool on Github, a pip package, and as a service through REST-API. [190]

ContrXT aims to explain how a classifier changes its predictions through time. We describe the five building blocks composing ContrXT, as in Figure3.1: (A) the two text classifiers, (B) their post hoc interpretation using global, rule-based surrogate models, (C) the Trace step, (D) the eXplain step and, finally, (E) the generation of the final explanations through indicators and Natural Language Explanations (NLE).

(A) Text classifiers. ContrXT takes as input two *text* classifiers $\psi_{1,2}$ on the same target class set C, and the corresponding training datasets $D_{1,2}$. As clarified in [250], classifying $\mathscr{D}_i$ under C consists of $|C|$ independent problems of classifying each $d \in \mathscr{D}_i$ under a class c_i for $i = 1, \ldots, |C|$. Hence, a *classifier* for c_i is a function $\psi : \mathscr{D} \times C \to \mathbb{B}$ approximating an unknown target function $\breve{\psi}$.

Output: Two black box classifiers on the same class set.

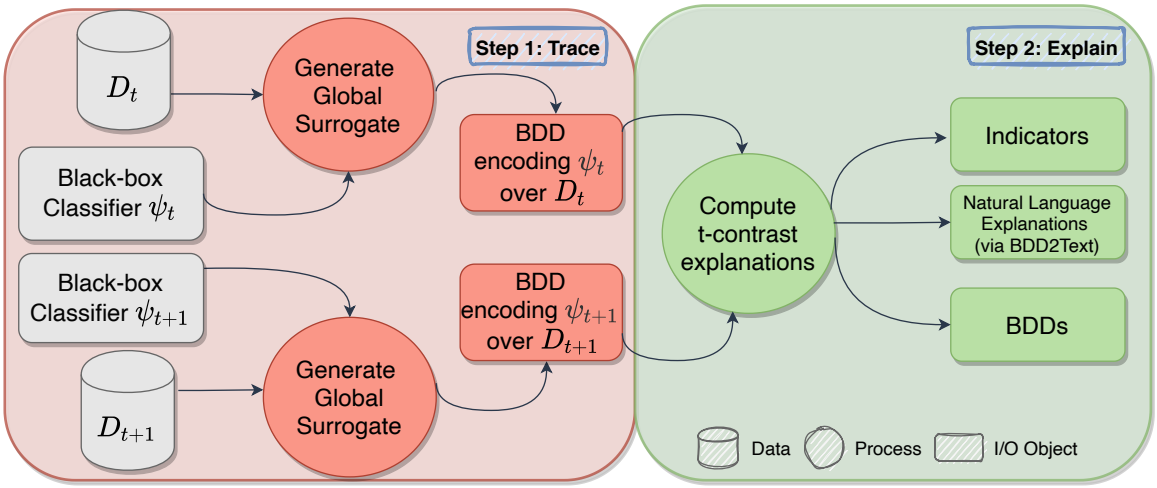


Figure 3.1 Overview of `ContrXT`*, taken from [188]*

(B) Post-hoc interpretation. Following the study about ML post-hoc explanation methods of [43], one of the approaches consists in explaining a black box model globally by approximating it to a suitable interpretable model (i.e., the *surrogate*) solving the following:

$$p_g* = \arg\max_{p_g \in I} \frac{1}{X} \sum_{x \in X} S(p_g(x), \psi(x)) \tag{3.1}$$

where I represents a set of possible white box models to be chosen as surrogates, and S is the fidelity of the surrogate p_g, which measures how well it fits the predictions of the black box model ψ. In addition to [43], `ContrXT` adds $\Omega(p_g) \leq \Gamma$ as a constraint to Eq. 3.1 to keep the surrogate simple enough to be understandable while maximising the fidelity score. The constraint measures the complexity of the model whilst Γ is a bounding parameter.[1] In the global case, the surrogate model p_g approximates ψ over the whole training set X taken from $\mathscr{D}$ which is representative of the distribution of the predictions of ψ.

Output: Two white box, rule-based surrogates $p_{1,2}$ of $\psi_{1,2}$

(C) Trace. This step aims at tracing the logic of the models $p_{1,2}$ while working on a datasets $D_{1,2}$.

It generates the classifiers' patterns through a global interpretable predictor (i.e., the surrogate), then it is encoded into the corresponding Binary Decision Diagram (BDD) [39].

[1] in case the surrogate is a decision tree, $\Omega(p_g)$ might be the number of leaf nodes whilst it could be the number of non-zero coefficients in case of a logistic regression

A BDD is a rooted, directed acyclic graph with one or two terminal nodes of out-degree zero, labelled 0 or 1. BDDs are usually reduced to canonical form, which means that given an identical ordering of input variables, equivalent Boolean functions will always reduce to the same BDD. Reduced ordered BDDs allow `ContrXT` to (i) compute compact representations of Boolean expressions, (ii) apply efficient algorithms for performing all kinds of logical operations, and (iii) guarantee that for any function $f : \mathbb{B}^n \to \mathbb{B}$ there is one BDD representing it, testing whether it is true or false in constant time.

Output: two BDDs $b_{1,2}$ representing the logic of $p_{g1,2}$.

(D) eXplain. This step takes as input the BDDs $b_{1,2}$, that formalises the logic of the surrogates $p_{g1,2}$, and computes the BDDs encoding the *differences* between the two. Step D manipulates the BDDs generated from the Trace step to explain how ψ_1 and ψ_2 differ (i) *quantitatively* by calculating the distance metric defined below (*aka*, Indicators), and (ii) *qualitatively* by generating the BDDs of the added/deleted patterns over multiple datasets D_{t_i}.

Definition 3.1.1 (T-contrast explanations through BDDs) *Given $f_1 : \mathbb{B}^n \to \mathbb{B}$ and $f_2 : \mathbb{B}^m \to \mathbb{B}$ we define:*

$$f_1 \oslash f_2 = \neg f_1 \wedge f_2 \qquad (3.2) \qquad f_1 \oeslash f_2 = f_1 \wedge \neg f_2 \qquad (3.3)$$

The goal of the operator $\oslash$ ($\oeslash$) is to obtain a boolean formula that is true iff a variables assignment that satisfies (falsifies) f_1 is falsified (satisfied) in f_2 given f_1 (f_2). Let b_1 and b_2 be two BDDs generated from f_1 and f_2 respectively, we synthesise the following BDDs:

$$b_{\oslash}^{b_1,b_2} = b_1 \oslash b_2 \qquad (3.4) \qquad b_{\oeslash}^{b_1,b_2} = b_1 \oeslash b_2 \qquad (3.5)$$

where $b_{\oslash}$ ($b_{\oeslash}$) is the BDD that encodes the reduced ordered classification paths that are falsified (satisfied) by b_1 and satisfied (falsified) by b_2. We also denote as

- *$var(b)$ the variables of b;*
- *$sat(b_{\oslash}^{b_1,b_2})$ all the true (satisfied) paths of $b_{\oslash}^{b_1,b_2}$ removing $var(b_1) \setminus var(b_2)$;*
- *$sat(b_{\oeslash}^{b_1,b_2})$ all the true (satisfied) paths of $b_{\oeslash}^{b_1,b_2}$ removing $var(b_2) \setminus var(b_1)$.*

Both $b_{\oslash}^{b_1,b_2}$ and $b_{\oeslash}^{b_1,b_2}$ encode the differences in the logic used by b_1 and b_2 in terms of feature presence (i.e., classification paths).

Indeed, $b_{\oslash}^{b_1,b_2}$ ($b_{\oeslash}^{b_1,b_2}$) can be queried to answer a T-contrast question like *"Why does a path on b_1 had a true (false) value, but now it is false (true) in b_2?"*. Clearly, features discarded (added) by b_2 are removed from paths of $b_{\oslash}^{b_1,b_2}$ ($b_{\oeslash}^{b_1,b_2}$) as they are used by ψ_1.

Output: Two BDDs $b_{\oslash}^{b_1,b_2}$ and $b_{\oeslash}^{b_1,b_2}$ encoding the rules used by b_2 but not by b_1 and vice-versa.

(E) Generation of final explanations: Starting from $b_{\oplus}^{b_1,b_2}$ and $b_{\ominus}^{b_1,b_2}$, the final explanations are provided through a set of *indicators* and *Natural Language Explanations.*

Indicators estimate the differences between the classification paths of the two BDDs through the Add and Del values (see Eq. 3.6 and 3.7). To compare *add* and *del* across classes, we compute the *Add_Global* (*Del_Global*) as the number of paths to true in $b_{\oplus}$ ($b_{\ominus}$) over the corresponding maximum among all the $b_{\oplus}^{c}$ ($b_{\ominus}^{c}$) with $c \in C$. In the case of a multiclass classifier, as for 20newsgroup, `ContrXT` suggests focusing on classes that changed more with respect the indicators distribution.

$$Add(b_{\oplus}^{b_1,b_2}) = \frac{|sat(b_{\oplus}^{b_1,b_2})|}{|sat(b_{\oplus}^{b_1,b_2})| + |sat(b_{\ominus}^{b_1,b_2})|} \tag{3.6}$$

$$Del(b_{\ominus}^{b_1,b_2}) = \frac{|sat(b_{\ominus}^{b_1,b_2})|}{|sat(b_{\oplus}^{b_1,b_2})| + |sat(b_{\ominus}^{b_1,b_2})|} \tag{3.7}$$

Natural Language Explanations (NLE) exhibits the added/deleted paths derived from $b_{\oplus}$ and $b_{\ominus}$ to final users through natural language. `ContrXT` uses the last four steps of *six NLG tasks* described by [89], responsible for *microplanning* and *realisation*. In our case, the structured output of BDDs obviates the necessity of *document planning* which is covered by the first two steps. The explanation is composed of two main parts, corresponding to Add and Del paths. Content of each part is generated by parsing the BDDs, extracting features, aggregating them using Frequent Itemsets technique [232] to reduce the redundancy, inserting the related parts in the predefined sentences [243].

3.2 Results on a Benchmark Dataset

Evaluation. `ContrXT` was evaluated in terms of approximation quality to the input model to be explained (i.e., the fidelity of the surrogate) on 20newsgroups, a well-established benchmark used in [135] to build a reproducible text classifier, and in [238], to evaluate LIME's effectiveness in providing local explanations. We ran `ContrXT` over different classifiers, trained through the most used algorithms, such as linear regression (LR), random forest (RF), support vector machines with RBF (SVM), Naive Bayes (NB), Bidirectional Gated Recurrent Unit (bi-GRU) [58], and BERT [73] *(bert-base-uncased)* with a sequence classification layer on top. Results are shown in Table 3.1. We considered and evaluated

ML Algo	Model F1-w D_{t_1}	Model F1-w D_{t_2}	Surrogate Fidelity F1-w D_{t_1}	Surrogate Fidelity F1-w D_{t_2}
LR	.88	.83	.76 (±.06)	.78 (±.07)
RF	.78	.74	.77 (±.06)	.79 (±.07)
SVM	.89	.84	.76 (±.06)	.78 (±.06)
NB	.91	.87	.76 (±.06)	.78 (±.06)
bi-GRU	.79	.70	.77 (±.06)	.78 (±.06)
BERT	.84	.72	.78 (±.05) •	.83 (±.06) •

Table 3.1 `ContrXT` *on 20newgroups (D_{t_1}, D_{t_2} from [135]) varying the ML algorithm. • indicates the best surrogate.*

all the global surrogate models surveyed by [43], representing state-of-the-art. Approaches falling outside the goal of `ContrXT` (e.g., SP-LIME [238] and k-LIME [111] whose outcome is limited to the feature importance values) and papers that did not provide the code were discarded.

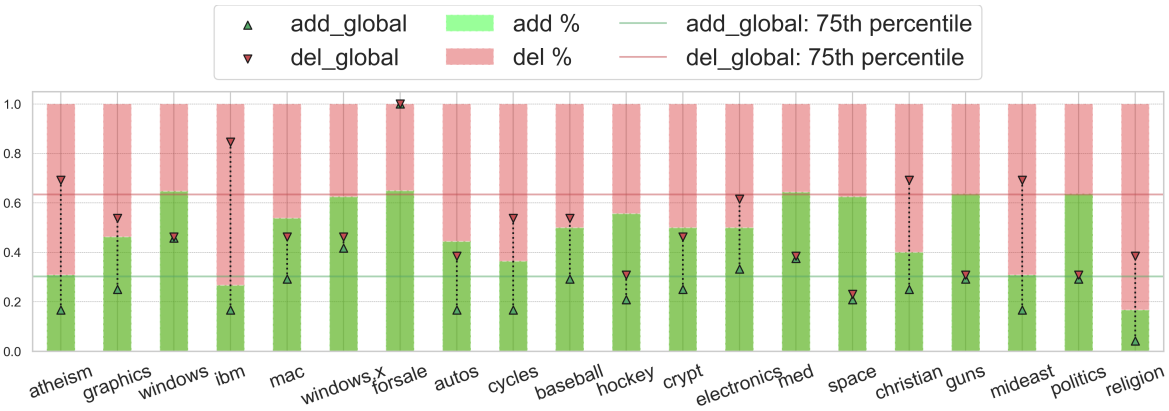


Figure 3.2 Indicators for the changes in classification paths from t_1 to t_2 for each 20newsgroup class. On the x-axis, we present the classification classes, and on the y-axis the ADD/DEL indicators

Results Comment for 20newsgroup. One might inspect how the classification changes from ψ_1 to ψ_2 for each class, i.e., which are the paths leading to class c at time t_1 (before) that lead to other classes at time t_2 (now) (*added paths*) and those who lead to c at t_2 that were leading to other classes at time t_1 (*deleted paths*). Focusing on the class *atheism* of

Figure 3.2 the number of deleted paths is higher than the added ones. Figure 3.3 reveals that the presence of the word *bill* leads the ψ_2 to assign the label *atheism* whilst the presence of such a feature was not a criterion for ψ_1. Conversely, ψ_1 used the feature *keith* to assign the label, whilst ψ_2 discarded this rule. Actually, both terms refer to the name of the posts' authors.

The example of Figure 3.3 sheds light on the goal of `ContrXT`, which is providing to the final user a way to investigate why ψ_2 classified documents to a different class with respect to ψ_1, as well as monitoring future changes. NLE allows the user to discover that -though the accuracy of ψ_1 and ψ_2 is high[2]- the underlying learning functions (i) learned terms that should have been discarded during the preprocessing, (ii) ψ_2 persists in relying on those terms, which are changed after retraining (using *bill* instead of *keith*), and (iii) having *political_atheist* is no longer enough to classify in the class.

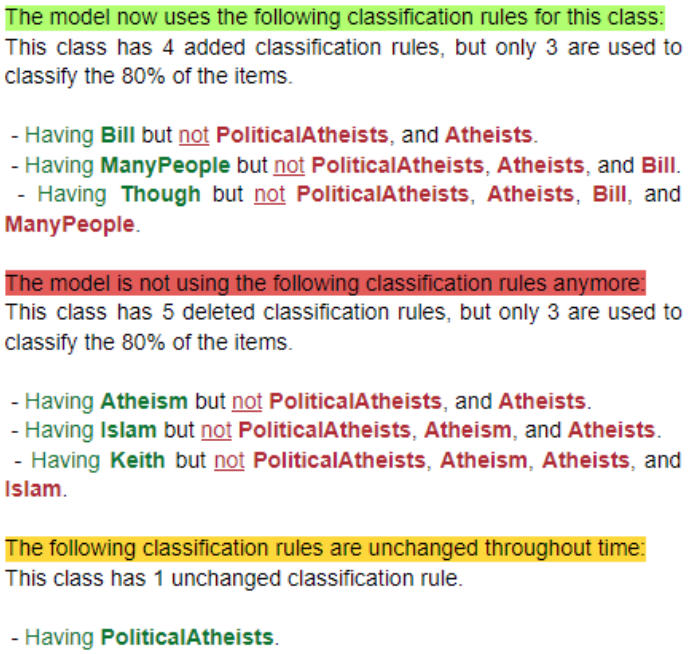


Figure 3.3 NLE for alt.atheism using the BERT model of Table 3.1

Evaluation through Human Subjects. We designed a study to assess if - and to what extent - final users can understand and describe what differs in the classifiers' behaviour by

[2]The Spearman correlation test revealed the accuracy is not correlated with the ADD/DEL indicators, confirming they provide additional insights beyond the quality of the trained models

looking at NLE outputs. We recruited 15 participants [3]from *prolific.co* [215] that were asked to look at NLE textual explanations and to select one (or more) statements according to the meaning they catch from NLEs. Results showed that the participants understood the NLE format and answered with an 89% accuracy on average,

and an F1-score of 87%. Finally, we computed Krippendorff's alpha coefficient to estimate the extent of agreement among users. We reached a value of 0.7, which [153] considers acceptable to positively assess the subjects' consensus.

Getting `ContrXT`. `ContrXT` can be used either as a pip Python package [189] or as a service through REST API. The API is written using Python and the Flask library [104] and can be invoked using a few lines of code.

A load testing has been performed using locust.io to measure the quality of service of the `ContrXT`'s API, adding a virtual user every 10 sec. Our architecture reached a throughput of 2.55 users per second. Beyond this value, while the API service keeps working, it puts additional requests into a queue.

Demo Video. Available at https://tinyurl.com/ContrXTIJCAI

[3]No limitation on country of origin, with Doctorate degree (PhD/other) as highest education level completed in one of the following subjects: Computer Science, Computing (IT), Engineering, Management, Mathematics, Physics

4

Conversational Explanations

In the previous two chapters, we demonstrated the role NL plays in XAI and how NL explanations can overcome issues like lack of comprehension by layperson audience and enhance the effectiveness of explanations in general.

In the current chapter, we discuss how the HITL paradigm improves a classic XAI system by replacing one-directional and one-size-fit-all explanations with conversational explanations that are contextual and by considering the human not only as the user but as "human-in-the-loop" who by giving feedbacks and challenging models decisions, improve the system as a whole (see [140]). Our proposed method is model-agnostic and can be "plugged" into any ML system by using humans, not mere users, active stakeholders who can contribute to the results. We further formalize and build a model which incorporates Information-seeking, as an atomic dialogue type [289], to the model proposed by [182], in order to address the need for more customisable XAI systems [259]. Figure 4.1 shows a classic ML-XAI system, together with our proposed method.

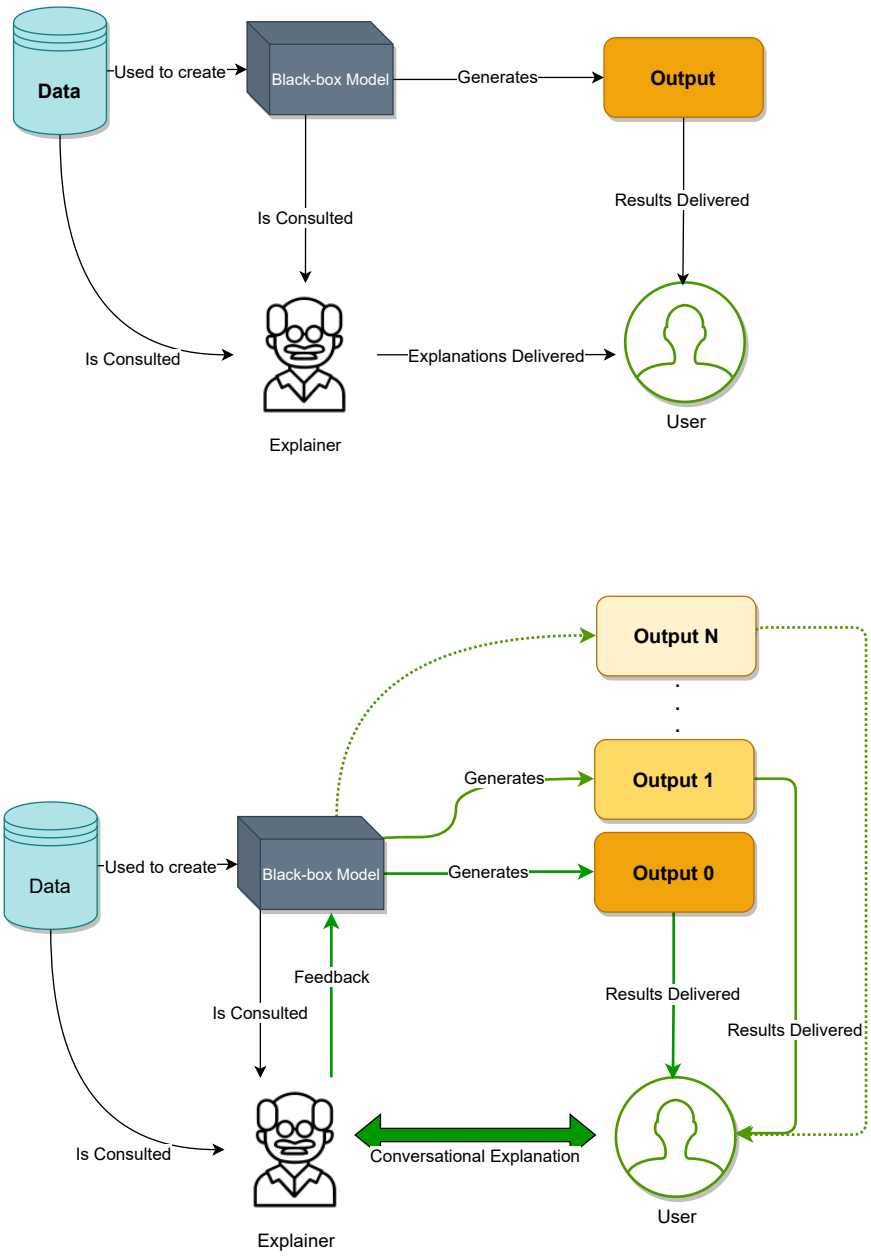


Figure 4.1 Above diagram: A classic ML-XAI system; Bottom diagram: ConvXAI, capable of generating conversational explanations in which the user contributes by challenging the model and giving feedback on the results. Such feedback then are used to suggest modifications of the model generation phase

4.0.1 Motivating Example.

To clarify the matter, let us consider an imaginary conversation among a user (👤) and a system (⚙) trained and deployed to classify million online job ads over a standard occupation taxonomy (see [92]).

👤: Why have you classified this job ad as *"Data engineer"*?

⚙: I saw *"Data"* and *"Architect"* words.

👤: What is the most important word used for classifying this job ad?

⚙: *"Architect"*

👤: Which word you didn't consider for the classification?

⚙: *"Senior"*

👤: What if the input was *"Senior Python data architect"*?

⚙: Then, the class would be *"Data Scientist"*.

👤: There is a label called "Data engineer". Why didn't you classify the input as this class?

⚙: The training data has only two records with this label. I don't have enough confidence to use it.

👤: Why instead of classifying it as *"Software Engineer"*, you have chosen *"Data engineer"*?

⚙: I never saw titles containing the word *"architect"* being labelled as *"Software engineer"* in the training data.

👤: Why you ignored the word *"senior"*?

⚙: I consider this word as *"too general"*.

👤: Answer me what "too general" means by considering me as a data scientist.

⚙: The TF-IDF score of the word "senior"- calculated based on the training data- is distant from the score of the rest of the terms.

👤: What if I tell you that I'm your developer?

⚙: In that case, I would say that I took that decision because the word senior is on the list of stopwords.

👤: Thanks! That's it.

⚙: Before you go...I've noticed that while the word *"senior"* is on the stopword list, the term *"experienced"* which has a high similarity to this word, is not.

To the best of our knowledge, no methods or frameworks that can satisfy all the points in this mock conversation exist, either in academia or in industry. To provide users with the most appropriate explanation, a system should be able to directly and dynamically interact with the user through NL. It should be able to access the black box, training data, and pipeline. Moreover, such a system should be able to identify and act upon users' knowledge and requirements, and warn them about possible abnormalities.

4.0.2 Contribution.

The contributions of this work are threefold:

- First, we formally extend the *conversational explanation framework* proposed by Madumal et al. [182] by introducing clarification dialogues as a separate dialogue type;
- Second, we contribute to the XAI community by bridging the gap between XAI and dialogue systems through an approach that allows the final user to interact with any state-of-the-art explainer to obtain both text and visual explanations from a black box model;
- Third, we implement our approach as an off-the-shelf Python tool, namely `ConvXAI`, which is publicly available to the community. We train the natural language understanding (NLU) model in `ConvXAI` through crowd-sourcing, and evaluate the system through a user study.

4.1 Problem Formulation

We build and formalise our framework on top of an existing model [182] which uses the modular agent dialogue framework (ADF) [195] to create a new atomic dialogue type [289]. Considering that the choice of explanation presentation impacts both the success of the conversation and the effectiveness of the explanation, we extend the model of Madumal et al. [182] by adding a new dialogue type, namely **Clarification**, which is closely related to the "information-seeking" dialogue type introduced by Walton and Krabbe [289], i.e. pursuing the goal of "acquiring or giving information". McBurney and Parsons [195] have described how this dialogue type becomes activated. While these Clarification acts have already been coded by Madumal et al. [182] in their return_question model, the mentioned locution covers

information about the explanandum, rather than the explanans itself, which often happens when a user does not have the required technical background to understand the presented explanans. Following Madumal et al. [182], the ADF is formalised as follows:

Definition 4.1.1 (Agent Dialogue Framework *ADF***)** *An ADF is a 5-tuple* = $ADF(\mathscr{A}, \mathscr{L}, \Pi_a, \Pi_c, \Pi)$ *where*

- $\mathscr{A}$ *is the set of agents* $\mathscr{A} = \{\mathscr{Q}, \mathscr{E}\}$*, with the labels* $\mathscr{Q}$ *and* $\mathscr{E}$ *denoting the Questioner (the explainee) and the Explainer, respectively;*
- $\mathscr{L}$ *is the set of logical representations about topics (denoted by p, q, r, ...);*
- Π_a *is the set of atomic dialogue types* $\Pi_a = \{G_E, G_A\}$*, where* G_E *is the explanation dialogue and* G_A *is the argumentation dialogue;*
- Π_c *is the set of control dialogues* $\Pi_c = \{Begin\_Question, Begin\_Explanation, Begin\_Argument, End\_Explanation, End\_Argument\}$*;*
- Π *is the closure of* $\Pi_a \cup \Pi_c$ *under the combination rule set.* Π *provides the set of formal explanation dialogues G.*

We extend the *ADF* proposed by Madumal et al. [182] towards an explanation dialogue model (EDM):

Definition 4.1.2 (Explanation Dialogue Model *EDM***)** *Let* $ADF = (\mathscr{A}, \mathscr{L}, \Pi_a, \Pi_c, \Pi)$ *be an ADF as in Def. 4.1.1. We define an EDM as the 5-tuple* $EDM = (\mathscr{A}, \mathscr{L}, \Pi'_a, \Pi'_c, \Pi)$ *where*

- $\Pi'_a = \Pi_a \cup G_C$*, in which the explanation dialogue* G_E *and the argumentation dialogue* G_A *are enriched with the clarification dialogue* G_C*;*
- $\Pi'_c = \Pi_c \cup \{Begin\_Clarification, End\_Clarification\}$.

McBurney and Parsons [195] presented an ADF in the form of a three-level hierarchy with the following layers:

Definition 4.1.3 (Topic Layer) *As the lowest level of the mentioned hierarchy, the topic layer presents the discussion topics, i.e. matters under discussion. Denoted by lowercase Roman letters p, q, r, ..., these topics can refer either to real-world objects or to the state of affairs [195].*

We extend the dialogue layer proposed by Madumal et al. [182] by adding a clarification (G_C) to the explanation (G_E) and argumentation (G_A):

Definition 4.1.4 (Dialogue Layer) *As the next level in the hierarchy, the dialogue layer models particular types of dialogues through four rules [182]:*

$$G = \{\Theta, \mathscr{R}, \mathscr{T}, \mathscr{C}\mathscr{F}\} \tag{4.1}$$

where Θ *(**Locutions**) denotes rules that determine which utterances are permitted in the dialogue-game,* $\mathscr{R}$ *(**Combination rules**) defines the dialogical context of the applicability of locutions,* $\mathscr{T}$ *(**Termination rules**) determines the ending of a dialogue, and* $\mathscr{C}\mathscr{F}$ *(**Commitments**) determines the circumstances in which players express commitment to a preposition.*

The dialogue layer consists of explanation (G_E*), argumentation (*G_A*), and clarification (*G_C*):*

$$G_E = \{\Theta_E, \mathscr{R}_E, \mathscr{T}_E, \mathscr{C}\mathscr{F}_E\} \tag{4.2}$$

$$G_A = \{\Theta_A, \mathscr{R}_A, \mathscr{T}_A, \mathscr{C}\mathscr{F}_A\} \tag{4.3}$$

$$G_C = \{\Theta_C, \mathscr{R}_C, \mathscr{T}_C, \mathscr{C}\mathscr{F}_C\} \tag{4.4}$$

with Θ_E*,* Θ_A*, and* Θ_C *defined as follows.*

Definition 4.1.5 (Legal Locutions) Θ_E = *(explain, affirm, further_explain, return_question)*
Θ_A = *(affirm_argument, counter_argument, further_explain)*
Θ_C =*(affirm_clarification, further_explain)*

Similar to Madumal et al. [182], we use Figure 4.2 to define the commencement rules (not present in the schema), combination rules (i.e. $\mathscr{R}_E$, $\mathscr{R}_A$, and $\mathscr{R}_C$), and termination rules (i.e. $\mathscr{T}_E$, $\mathscr{T}_A$, and $\mathscr{T}_C$) via a transition state diagram constructed using the Unified Modelling Language (UML) notation.

Definition 4.1.6 (Control Layer) *State transitions lead to shifts from one type of dialogue to another within a complex dialogue (see Definition 4.1.7).*

Definition 4.1.7 (Complex Dialogue) *An extended sequence of dialogue where there is a shift between dialogue types (see [195]).*

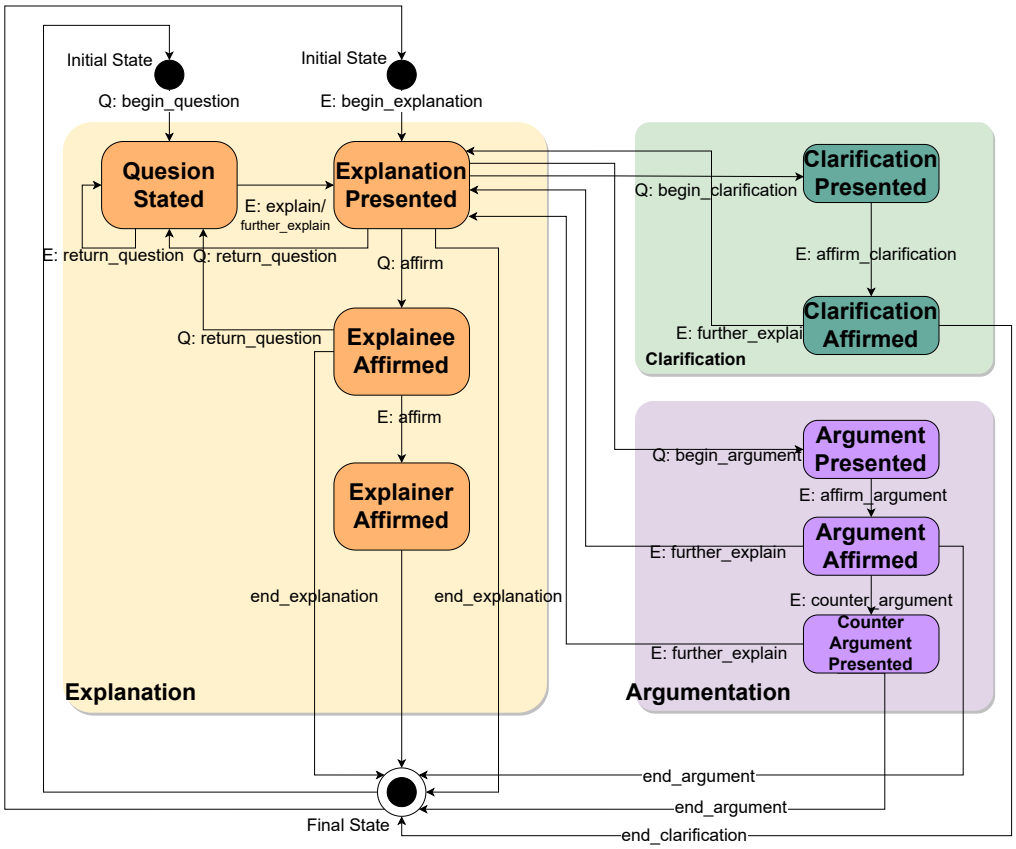


Figure 4.2 UML state transition diagram of the proposed extended Explanation Dialogue Model that enhances the work of [182] through clarification dialogue type. Locutions starting with Q and E refer to the Questioner and the Explainer, respectively.

To facilitate the comprehension of the provided definitions and the state transition diagram (see Figure 4.2), we provide the following conversation between a human and an agent regarding attrition prediction in a specific organisation. Note that this conversation represents a complex dialogue (see Definition 4.1.7) consisting of explanation, argumentation, and clarification dialogue types.

: [Begin_Question] I see that the predicted attrition percentage for some departments is relatively low; can you confirm this?

: [Begin_Explanation] [Explain] Yes, about 18% of departments have an attrition rate of less than 5%.

: [Return_Question] Where do the training data come from?

: [Further_Explain] The model was trained based on employees' activity and demographic records for the past five years, except for the procurement department, which has some missing data in 2020.

: [Begin_Argument] How did you handle those missing data?

: [Affirm_Argument] [Further_Explain] Where possible, I used interpolation to fill them.

: [Begin_Clarification] What is interpolation?

: [Affirm_Clarification] [Further_Explain] [End_Clarification] Simply put, it means filling the missing data using the known data, for example, using the average value of the available data.

4.2 How ConvXAI works

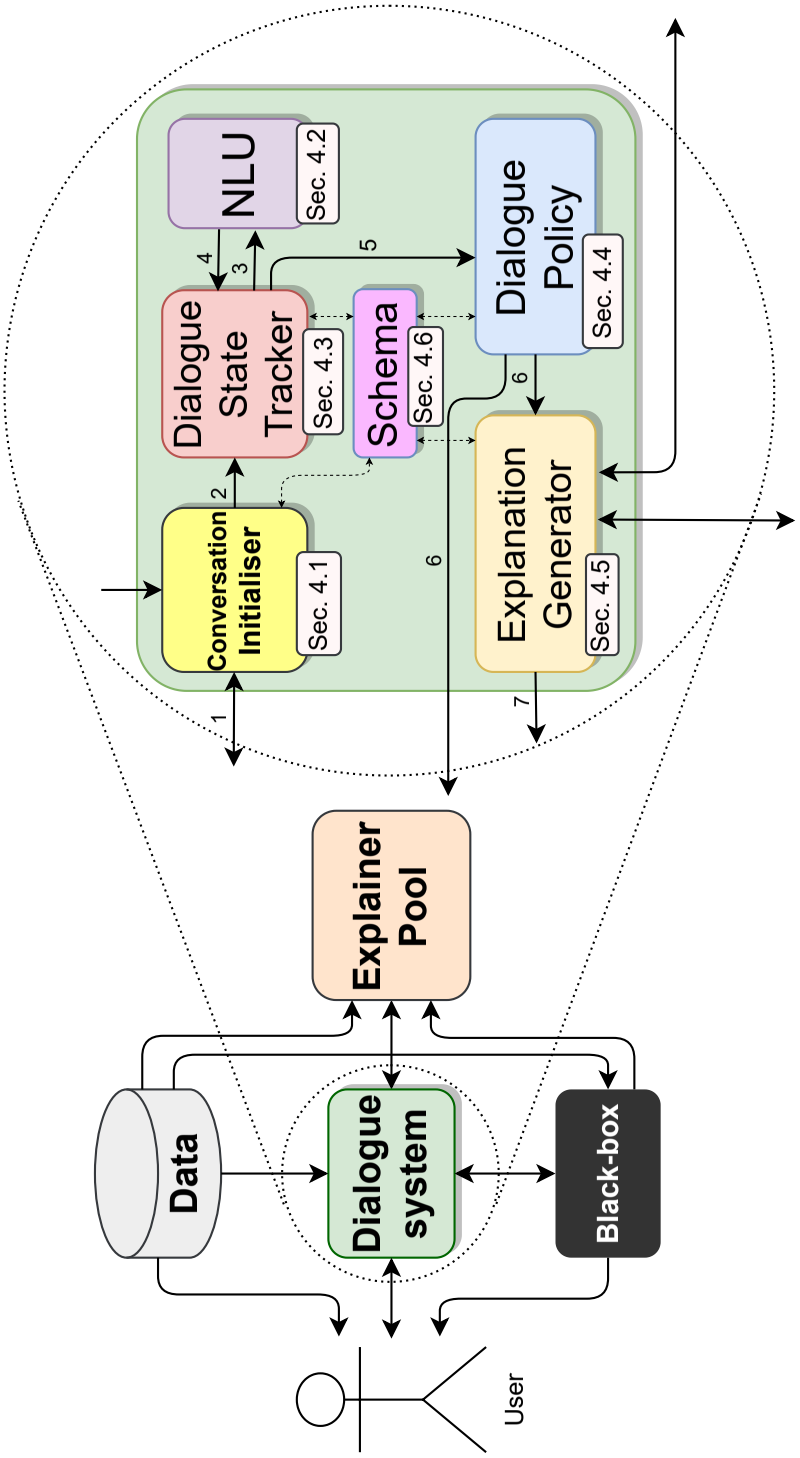


Figure 4.3 Components of ConvXAI *- Numbers on arrows show the sequence of actions: 1)user interacts with conversation initialiser 2)conversation initialiser communicates with DST 3)DST uses NLU 4)NLU passes its output to DST 5)DST passes the sate to DP 6) DP either directly answers the user or communicate with explanation generator 7)explanation generator interacts with the user*

`ConvXAI` is built upon three pivotal components: the data, the black box, and the explainer. On top of these components, we add a dialogue system that will be described in detail in the following sections. While we are not the first authors to incorporate a dialogue module in an XAI system (e.g. [115, 6, 217, 243]), to the best of our knowledge, we are the first to propose an entirely component-agnostic multimodal conversational XAI system. Our proposed `ConvXAI` system transforms the static/generic explanations into a two-way communication in which the dialogue is tailored based on the user's profile. Figure 4.3 depicts various components of `ConvXAI`, while Procedure 1 presents the algorithm for conversational explanations. This algorithm shows how, starting from the Schema (see Section 4.2.6), the agent carries out a conversation with the goal of providing the user with an explanation.

`ConvXAI` utilises both machine learning (ML) and rule-based techniques inside the dialogue system. Although the dialogue state tracker and DP both use rules to perform their tasks, they do so by using the Schema component (see Section 4.2.6) instead of having rules to handle singular cases. This design reduces the time and effort that a classic rule-based system requires to be created from scratch or adapted to a new environment.

Procedure 1 Explanation Dialogue

Require: Schema

$State \leftarrow \emptyset; Answer \leftarrow \emptyset; Initial\_Data \leftarrow \emptyset; current\_intent \leftarrow \emptyset$
while $Initial\_Data = \emptyset$ **do**
    $Initial\_Data \leftarrow user\_input$
$compatible\_explainers \leftarrow Schema(initial\_data)$ //Schema as in Section 4.2.6
$user\_turn \leftarrow user\_input$
$State \leftarrow DST(user\_turn)$ //DST as in Procedure 3
$Answer \leftarrow DP(State)$ //DP as in Procedure 4
return *Answer*

4.2.1 Conversation Initialiser

The first component of `ConvXAI` interacts with user queries through the Conversation Initialiser (CI), which obtains the initial data from the user, including the dataset, black box model, and user profile. Upon inserting this information from the user through predefined options (e.g. datasets and models already integrated in the system), the CI performs a series of checks to ensure the compatibility of the inputs and that at least one explainer is able to

create an explanation for the chosen user profile (see Section 4.2.5 for more information about the connection of the user profile and presentation method). If this criterion is met, after training the selected model using the chosen dataset, the CI provides the user with the results, i.e. the outcome of the black box model, as a downloadable Excel file. If this criterion is not met, i.e. no explainer is found for this particular set of inputs, the CI informs the user of this by presenting a predefined message.

4.2.2 Natural Language Understanding

In our work, we use the NLU model of the RASA[1] framework as a joint classifier, which is the Python implementation of the DIET model proposed by Bunk et al. [42]. This model uses conditional random fields [156] and transformers [279] to extract entities and classify the general intent of the user input. The following explains the reason for choosing DIET as the NLU model of `ConvXAI`:

Performance: As reported by Bunk et al. [42], DIET outperforms state-of-the-art models such as HERMIT [278] or the fine-tuned BERT on both intent classification and entity extraction sub-tasks using the NLU benchmark dataset [174]. By comparing DIET to several other NLU services (e.g. Microsoft's LUIS[2]), Braun et al. [37] showed that, while DIET produced the second-best performance overall, it outperformed other NLU services on the *Chat* dataset while producing similar performance on the *webapp* and *Ask Ubuntu* datasets[3]. Considering only intent classification, Lorenc et al. [176] compared DIET with several public NLU services (e.g. Google's DialogFlow[4]) and standard classification algorithms (e.g. Support Vector Machine (SVM) [66]) on the datasets used by Braun et al. [37], and found that DIET achieved F1-scores that were higher than or similar to other methods. **Open source**: As mentioned by Braun et al. [37], commercial services are fed data by hundreds, if not thousands, of users every day, resulting in better performance. Being used in the RASA NLU platform, DIET profits from the benefits of numerous users, has been released publicly, and is subject to continuous improvements, resulting in gradual improvements

[1] https://rasa.com/
[2] https://www.luis.ai/
[3] All datasets can be accessed from https://github.com/sebischair/NLU-Evaluation-Corpora
[4] https://cloud.google.com/dialogflow

over time. **Architecture flexibility**: Apart from its easy-to-modify interface regarding the hyperparameters or components used for tokenisation and featurisation, DIET benefits from a flexible architecture that makes it possible to work both with and without pre-trained language models. Depending on the use case, this can be highly beneficial. As the NLU model, DIET extracts the intent and entities from user input, while considering the predefined reliability threshold under which the user will be asked to rephrase their input. Procedure 2 describes this process.

To train the mentioned implementation of the DIET model, the training phrases and their intents should be prepared in a specific format. Although preparing data with this format is relatively undemanding for a single dataset, `ConvXAI` will not work with any dataset provided by the user. To overcome this problem, we have created a process that automatically generates the training data using a delexicalisation procedure.The following is an example of an intent and its entities represented using four phrases:

Intent: what_if_add
Example:
- What if I add ["*others*"](token) to [instance 69](record)?
- What would be the output if I add ["*giving*"](token) to [instance 42](record)?
- How will adding ["*people*"](token) change the output for [instance 120](record)?
- What is the impact of adding ["*brought*"](token) to [instance 116](record)?

Delexicalisation

This procedure consists of the following steps: intent discovery, entity discovery, template creation, and training data generation.
In the first two steps, we discover a group of highly used intents and the entities in the corresponding user queries. The latter two steps use this information and generate the training data.
Intent discovery: Given the supervised nature of the NLU model, intents should be defined before being used to train the model. To do so, we started with the 12 intents (query types) defined by Kuzba [155], and revised them for our work by adding 12 new intents and

removing six unrelated ones through a crowd-sourcing activity[5]. We hired ten participants from the *prolific.co* [215] platform, and asked each to write down all possible questions (at least six unique ones) that they could ask about three datasets/tasks described in the study. From the 212 questions that the participants generated, we removed the duplicated and unrelated questions, which left us with 53 unique questions. We then manually categorised these questions into 17 intent categories (see Table 4.1). Finally, the following intents were added, which address the questions that occur naturally in human conversation: *greet, goodbye, affirm, deny*. An additional intent called *bot_challenge*, which refers to users wanting to talk with a human instead of a chatbot, was also added.

[5]https://forms.gle/JArrvVVYJR1jdpUs8

Table 4.1 Final list of explanation-related intents

Scope	Model type	Intent	Description	Example
local	regression	why_not_regression	User was expecting to see a specific outcome	Why such a low value is predicted for instance 45?
local	classification	why_not	User was expecting to see a specific outcome	Why record 12 is not classified as Spam?
local	Any	expectation_not_met	User was expecting to see another outcome	I was expecting to see record 32 being classified as "Setosa"
global	Any	overall_contribution_to_label	How features are contributing to a specific outcome	What is the contribution of features in predicting label A?
global	Any	feature_importance_global	How features are contributing to outcomes	How the model is using features to come up with results in general?
local	Any	why_this	The reason for seeing a specific outcome	Why such an outcome is predicted for record number 2?
local	Any	single_contribution_to_label	The reason for seeing a specific outcome	Why I'm seeing such output for instance 1 ?
local	Any	most_important_feature	The reason for seeing a specific outcome	Which are the most important attributes for the outcome of instance 36?
local	Any	list_features	The reason for seeing a specific outcome	Give me a list of features and their impact for predicting record 5
local	classification	what_if_i_del	Effect of removing a word	What if I remove the word "extra" from the last record?
local	classification	what_if_i_add	Effect of adding a word	What would be the outcome of the record 14 if I add the word "official"?
global	regression	feature_effect_regression	How features are contributing to outcomes	How the model is using input data to predict the shown outcomes?
global	classification	feature_effect	How features are contributing to outcomes	Tell me how different features are being used to classify records
local	Any	what_if_i_subs	Effect of altering a feature	What if I change the word "man" to "woman" i the record 40?
local	Any	least_important_feature	Least important features used to generate a specific outcome	What features are used the least for prediction of instance 10?
global	classification	why_this_by_feature_category	feature importance for predictions of a specific label	What leads to predict a records as label B?
local	classification	why_not_this_record	User expect to see the same result for another record	Why the model didn't classify the record 4 similar to record 5?

Entity discovery: Next, we manually identified all entities from the generated questions as follows: *record, label, act_label, des_label, feature, new_val, token.*

Creating templates: Upon their identification, entities were replaced with the general name of the entity itself. For instance, the raw question "Why does the model predict record 13 as Angry?" would be transformed to "Why does the model predict **Instance** as **ActualLabel**?"

Generating training data: After generating templates, the placeholders were filled with the entities from the specific dataset, following the required format of the RASA framework. For example, the template phrase "Why does the model predict **Instance** as **ActualLabel**?" would be transformed to "Why does the model predict **[instance 68](Instance)** as **[Happy](ActualLabel)**?"

Procedure 2 NLU

Require: Question as in Section 2; $\tau \leftarrow reliability\ threshold$

1: $Intents \leftarrow \{\}; Entities \leftarrow \{\};$
2: $Intents, Entities \leftarrow NLU\_model(Question);$
3: $intent\_reliabilities \leftarrow Intents[reliability]$
4: $1 - best\_reliability \leftarrow intent\_reliabilities[best\_reliability]$
5: **if** $1 - best\_reliability < \tau$ *OR* $1 - best\_reliability = Unreliable$ **then**
6: $\quad Intent \leftarrow \emptyset$
7: **else**
8: $\quad Intent \leftarrow Max(Intent\_reliability)$
9: **return** $Intent, Entities$

4.2.3 Dialogue State Tracking

Given the component-agnostic architecture of `ConvXAI`, we found the rule-based method to be the most suitable for DST because it does not require an intensive amount of labelled data and is easier to debug and maintain. The rule-based method often has the limitation of following only the 1-best identified intent. Similar to Wang and Lemon [295], we address this limitation by training a logistic regression model for each user profile (see Section 4.3.2). This model classifies the 1-best intent identified by the NLU model as reliable or unreliable. Ten participants, corresponding to each user profile, formulated each identified intent (see Section 4.2.2) as alternative questions. We fed each question to the NLU model, which returned an array of possible intents with their reliability scores. We then manually labelled the 1-best intent as True/False. In the next step, we used the mentioned Boolean values, the

standard deviation among reliability scores for each question, and the user profile to train the binary classifier model described earlier. We assumed that the standard deviation value was a valid predictor of the model's difficulty in determining the correct intent. Each time a user input is passed to the NLU model, if any of the following conditions exist, an explicit confirmation [139] is used to validate the identified intent:

- The reliability score of the 1-best intent is lower than a predefined threshold (e.g. 80%);
- The logistic regression marks an intent as unreliable. In case of failure of the explicit confirmation, the user will be asked to reformulate their question.

If either of these conditions is true, following the basic concept of active learning [254], the user query is added to a separate dataset that is used to retrain the NLU model. The retraining procedure is activated if, for a specific user profile, the percentage of failed confirmations exceeds a predefined system threshold (e.g. 5%).

After confirming that the user has not changed their previous intent (i.e. changed their mind), `ConvXAI` refers to the Schema to return the state of the conversation. This process is described in Procedure 3.

Procedure 3 Dialogue State Tracking

Require: current_intent ; threshold; Schema

$Intent \leftarrow \emptyset; mind\_change \leftarrow False$
$Intent, Entities \leftarrow NLU(user\_input)$ //NLU as in Procedure 2
if $Intent \neq \emptyset$ **then**
  if $current\ intent = \emptyset$ **then**
    $current\ intent \leftarrow Intent$
  else
    $mind\_change \leftarrow True$
if $mind\_change = True$ **then**
  $State \leftarrow restart$
else
  $State \leftarrow Schema(initial\_data, user\_turn, current\_intent, Entities)$ //Schema as in Section 4.2.6
return State

4.2.4 Dialogue Policy

Considering `ConvXAI`, the DP module does not predict the best action; instead, it extracts the best action by considering the current state of the conversation and the given context. If

there is more than one alternative for the explainer and explanation presentation, a single instance is selected at random. To choose an adequate explanation, the DP consults the list of all available explainers prepared by the dialogue initialiser module (see Section 4.2.1) at the beginning of the conversation.

Procedure 4 shows how the DP unit determines the agent's turn (Answer) by checking for the type of input the user has provided.

Procedure 4 Dialogue Policy

Require: $State; Schema; compatible\_explainers$
1: $canned\_states \leftarrow Schema[canned\_states]$
2: **if** $State \in canned\_states$ **then**
3: $\quad Answer \leftarrow Schema[canned\_states][State]$ //Schema as in Section 4.2.6
4: **else**
5: $\quad explainer \leftarrow compatible\_explainers[0]$
6: $\quad Answer \leftarrow explainer(Schema)$
7: **return** Answer

4.2.5 Explanation Generator

One of the less-explored aspects of XAI is the presentation layer, where the explanations made by the explainer are transmitted to the end-user.
The choice of the representation method depends on several factors, such as the ease of producing the representation, availability of out-of-the-box solutions, perceived level of representation comprehension by users, and characteristics of the input data. The majority of presentation techniques used in the XAI literature fall into one of the following categories:

Graphics/plots These are the most popular methods in the literature. In our opinion, their popularity is rooted in the presence of appropriate tools and the relative simplicity of generating such graphics. This group mainly consists of *bar plots*, *line plots*, *trees*, *heatmap plots*, *histograms*, *scatter plots*, and *bubble plots* [238, 3, 252, 275].

Text This group contains methods that use text as their basis. Note that this does not mean the output is necessarily expressed in NL, but indicates that the main message is conveyed

through text rather than other techniques. The main textual representations are *rules*, *word annotations*, and *NL* text [13, 260, 53].

Images Image-based presentations are considered more sophisticated than plots/graphics, yet are more limited because they can only be applied if the target input is an image. The main types of this category are *image heatmaps*, *saliency masking*, and *image manipulation* [239, 19].

Reports Although this family is similar to the *text* category, reports have a more structured approach and are often combined with other methods (e.g. graphics). The main techniques in this category are *tabular reports*, *decision tables*, and *graphical table reports* [117, 281].

Although the majority of current XAI methods utilise one of the presentation methods described above, few combine different modes of presentation [123]. The component-agnosticism of `ConvXAI`means that it is not limited to a specific explainer method and, as a result, can provide its outputs in various modalities. Nevertheless, to enable non-technical users to use certain explanations that could be challenging in their initial forms (e.g. decision rules), a template-based natural language generation method is developed to transform these specific explanations into NL forms. To determine which presentation method is most suitable for each type of audience, we conducted a user study by showing different presentation alternatives of the same explanations to the users and evaluating the precision, understandability, usefulness, and complexity of each presentation (see Section 4.3) [99].

Upon deciding which kind of explanation, presentation, and explainer should be used to answer the user's question, the DP sends a request to the explanation generator (EG) unit with the information mentioned above. At this stage, depending on the presentation method (graphical, NL) and the explainers' characteristics, the EG either directly assesses the explainer's output to the user or modifies it based on the user's profile. For instance, if the generated explanation is in NL, but its initial form makes comprehension by a non-technical user difficult, the explanation goes through a parser and is transformed by a predefined template to a more user-friendly form.

4.2.6 Schema

The Schema is a JavaScript Object Notation structure which stores all the necessary information regarding integrated datasets, models, and user types. It also keeps a record of the user input in each dialogue. The Schema has two main tasks: first, it acts as the primary component of DST to ensure that the user provides all the required information (i.e. all the mandatory entities for a specific intent); second, it serves as an auxiliary to the DP and DST components, providing them with the legal locutions in each dialogue turn (see Definition 4.1.4).

4.2.7 Summary of `ConvXAI` tool

`ConvXAI` is a Python tool with a component-agnostic architecture that can customise the provided explanations based on user knowledge/experience and offers a dialogical interface for state-of-the-art explainers. `ConvXAI` relies on external explainers to fill the potential gap between the available explanations and what users consider "comprehensible". A specific type of dialogue, namely clarification dialogue, is used to ensure the usability of the outputs. Note that `ConvXAI`, as an open-source tool, can be extended by the community to include any explainer. The current explainers integrated into the tool are as follows:

- **LIME** [238]: plain-fact explanations, i.e. explanations which answer *what* questions, local explanations, i.e. explaining a single record instead of the model's output, through graphical and textual presentations.
- **SHAP** [179]: plain-fact, local, and global explanations through graphical and textual presentations.
- **FoilTree** [277]: Contrastive, i.e. explanations which answer *why* questions, through textual presentation.

4.2.8 Tool and User Interface

Using Node.js as the front-end and the Python language as the background, we created a chatbot application that can interact with users, obtain the necessary information to initialise the conversation, return the output of the black box model, and provide NL and visual explanations. Figures 4.4 to 4.9 present some screenshots of the application. All components

Figure 4.4 Example of Conversation Initializer

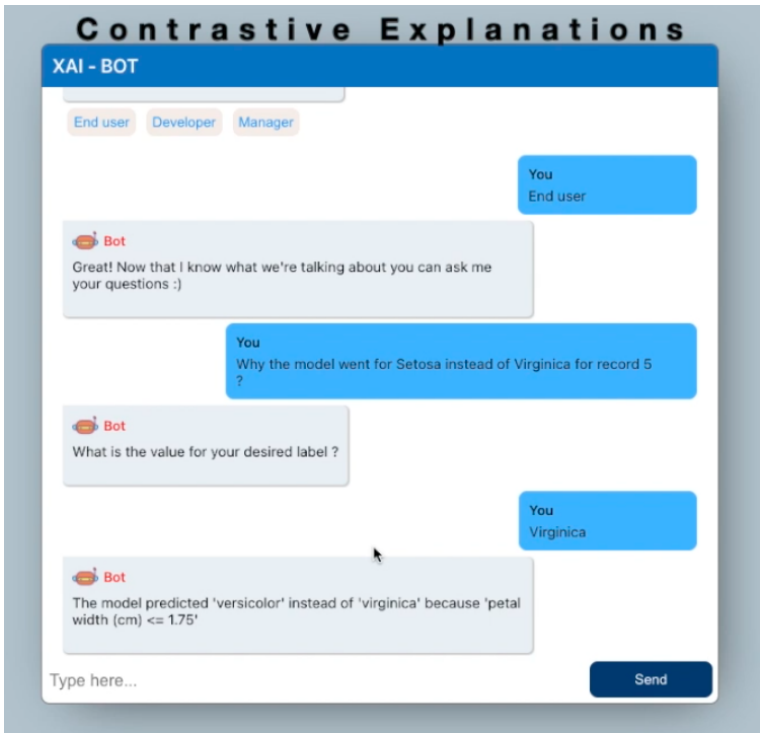


Figure 4.5 Example of a contrastive explanatin

Figure 4.6 Example of a global explanation

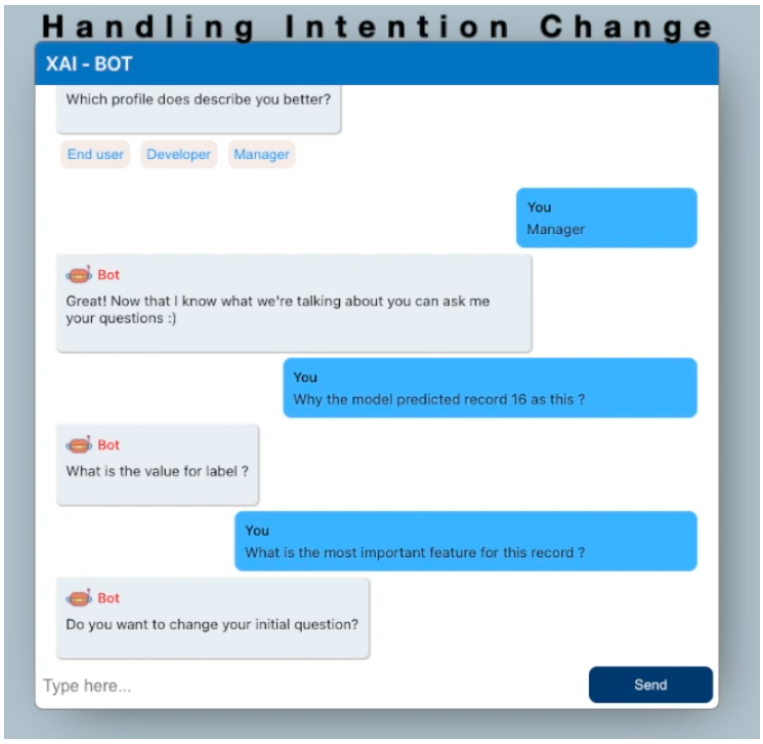


Figure 4.7 Example of how `ConvXAI` *handles user's intention change*

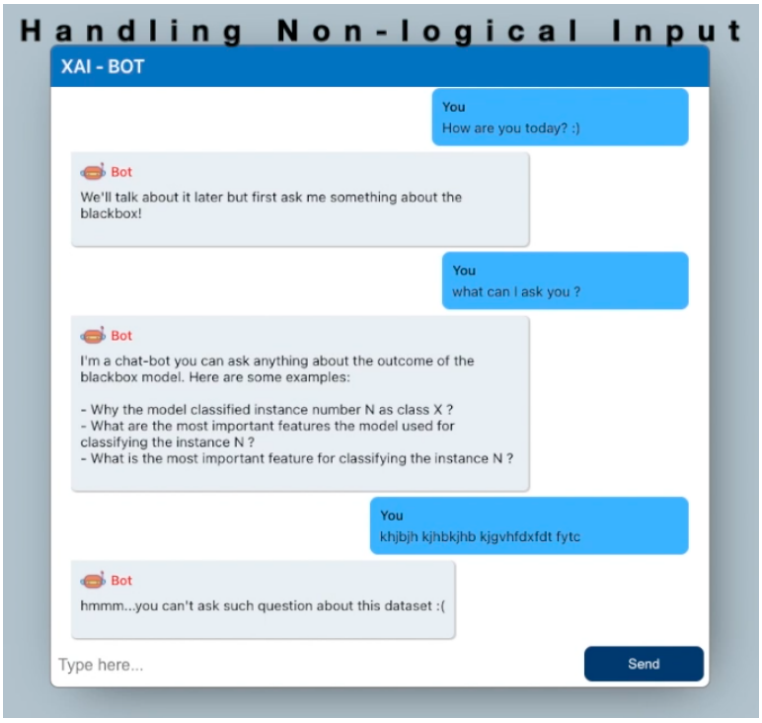


Figure 4.8 Example of how `ConvXAI` *handles user's non-logical input*

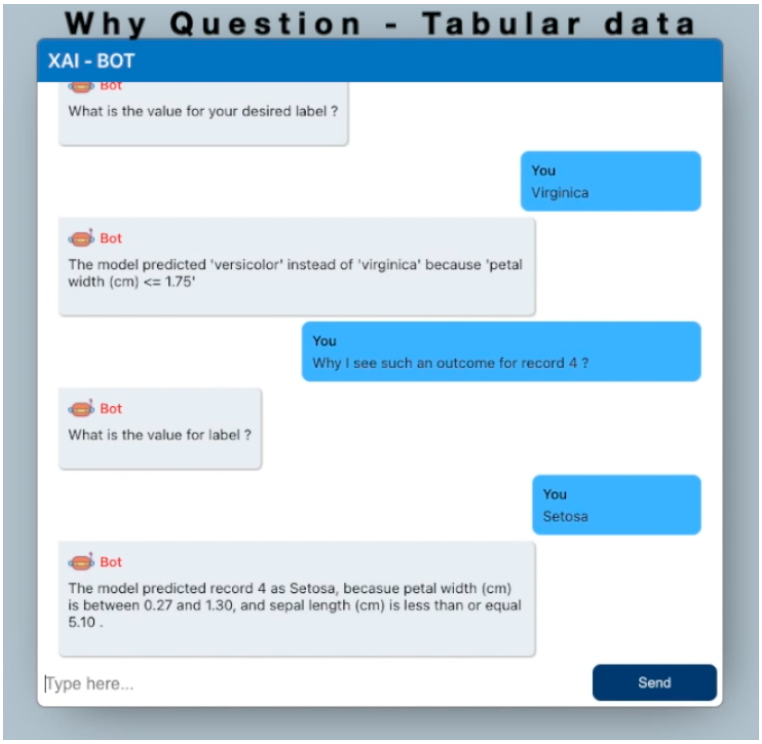


Figure 4.9 Example of a why explanation

demonstrated in Figure 4.3 are implemented in Python as separate modules, which facilitate the integration of new black box models and datasets with the current ones. While the components described in the previous sections create the backbone of the tool, there are several auxiliary functions which improve the user experience. These functions are described as follows.

Auxiliary functions

Compatibility Check Given the agnostic architecture of `ConvXAI`, there is no *a priori* check of compatibility among integrated components, i.e. there is not guaranteed to be a match among each tuple of the dataset, black box, and explainer. Thus, each time the user chooses these components, a compatibility check is performed to ensure that the black box can handle the specific task related to the dataset (e.g. text classification) and that there is at least one explainer model which can create at least one type of explanation (i.e. presentation method) adequate for the profile chosen by the user.

Topic Alteration While changing the argument topic is a natural process in human conversation, such an event can impact the performance of the DST unit in tracking the dialogue's state. To reduce this risk, each user turn [139] passes through the NLU model to ensure that (i) the user is not communicating a new intent and (ii) the user turn is similar to those related to the same intent and user profile. If such a control fails, the user will be asked whether they changed their idea and no longer wish to proceed with the initial question.

4.3 Evaluating `ConvXAI` through a User Study

4.3.1 Research Method

The usefulness and effectiveness of explanations provided by `ConvXAI` were evaluated through an online user study. The participants were divided into three groups based on their relationship with the technology and their managerial experiences, and asked to fill out an online form. Each participant was presented with four dialogues. At the end of each one, they were asked to evaluate the responses provided by the agent (explanation) and the eventual

clarifications, and assign an overall score to the dialogue. In the following section, we discuss the details of this experiment.

4.3.2 Experimental Design

Our study consists of four dialogues in which the agent and user argue about the results of an ML model constructed using a particular dataset and performing a specific task: **Iris** (multi-class classification), **Homonyms** (binary classification), **Boston housing** (regression), and **Breast cancer** (binary classification). Each case provides a description of the data, a sample of the data, and details about the task.

Participants Forty-five participants were randomly selected (balanced gender) from a research pool, with 15 participants in each of the non-technical, manager, and technical groups. To ensure the participants were fluent in English, besides asking for explicit confirmation, we selected them from the following countries: US, UK, Germany, France, Spain, Finland, Italy, Netherlands, Norway, Portugal, Sweden, and Switzerland. Group 1 [Non-Technical]: Low usage of technology; participants who either do not use technology at work or their usage is limited to two or three times a week; Group 2: [Managers] Junior, middle, or upper managers; and Group 3: [Technical] Tech-savvy participants; those who use technology at work daily and have at least graduate-level education.

Measures To evaluate the conversations, participants answered three categories of questions: (i) **Explanation Evaluation**: Statements that assess how easily the users understood the explanations without external help, their precision and granularity, usefulness regarding causality discovery, comprehension ease, and their usability (see Figures 4.10 to 4.15 for graphical explanations):

- I found that the explanation included all relevant known causal factors with sufficient precision and granularity;
- I did not need support to understand the explanation;
- I found the explanation helped me to understand the causality;
- I was able to understand the explanation with my knowledge base;
- I think that most people would learn to understand the explanation very quickly.

Table 4.2 Macro-structure of dialogues used in the user study

	Dataset	Explainer	Textual Explanation	Graphical Explanation	Clarification
Dialogue 1	Iris	Lime	Provided	Explanation 6	Not provided
Dialogue 2	Homonyms	Lime	Provided	Explanation 3	Provided
Dialogue 3	Boston Housing	SHAP	Not provided	Explanation 4 and 5	Provided for explanation 4
Dialogue 4	Breast Cancer	SHAP	Not provided	Explanation 1 and 2	Provided

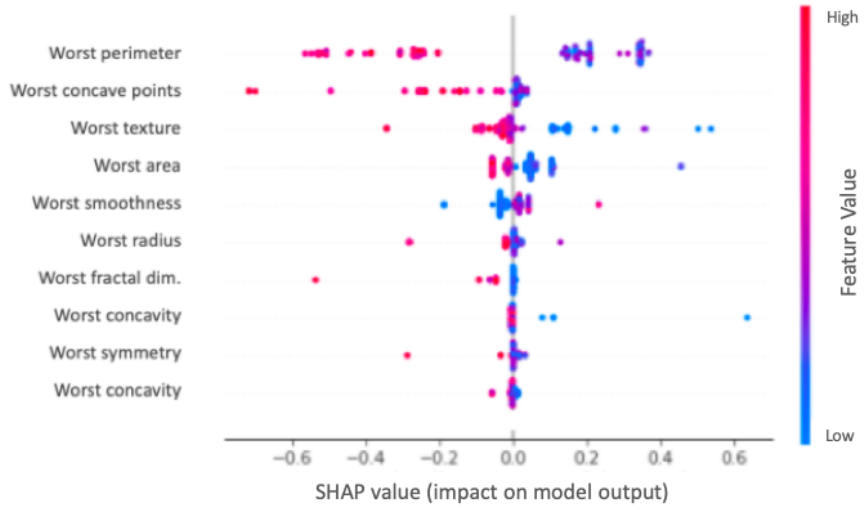


Figure 4.10 User Study: SHAP - Plot 4

(ii) **Clarification Evaluation**: Usefulness of the provided clarifications; and (iii) **Dialogue Evaluation**: To evaluate the overall performance of the conversations, we asked participants to state the extent to which the agent was able to understand their specific needs and provide the required explanations and clarifications. Table 4.2 provides a high-level description of the dialogues used in the user study.

Materials To conduct the online user study, we used *prolific.co* [215], a platform for online subject recruitment which explicitly caters to researchers and engages reviewers in performing research evaluations and surveys. Each subject was compensated £7.50 per hour.

4.3.3 Result Comments

Explanation evaluation: We studied how different groups of users perceived explanations and to what extent groups agreed or disagreed about various aspects of the explanations. Note that by doing so, unlike for the clarification evaluation, we are not directly evaluating

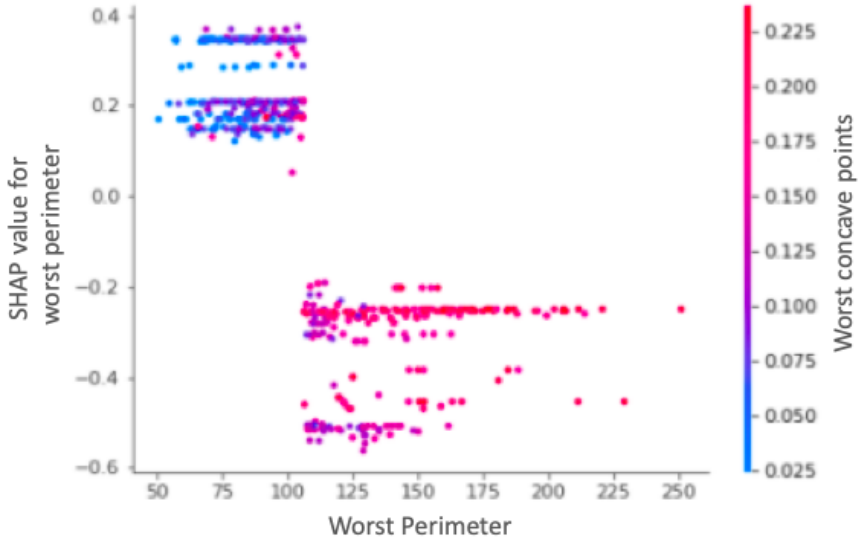


Figure 4.11 User Study: SHAP - Plot 3

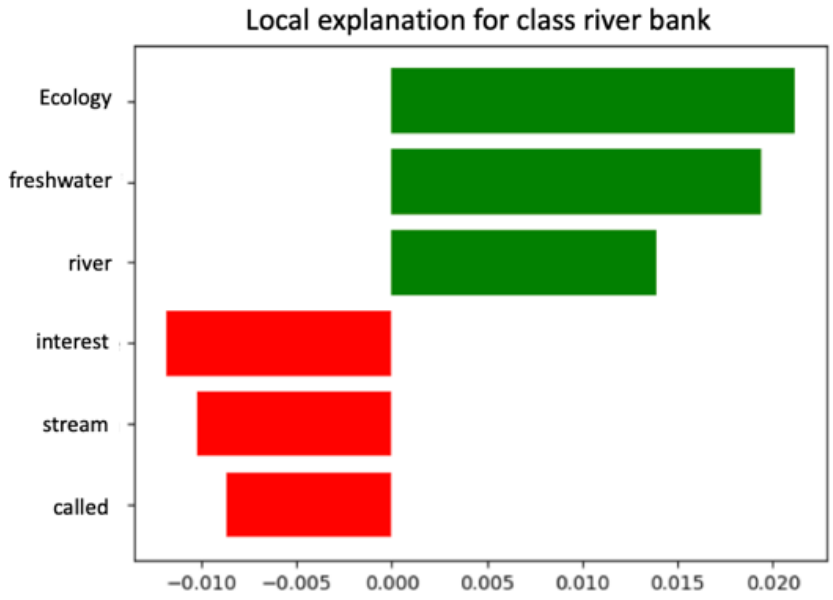


Figure 4.12 User Study: LIME - Plot 2

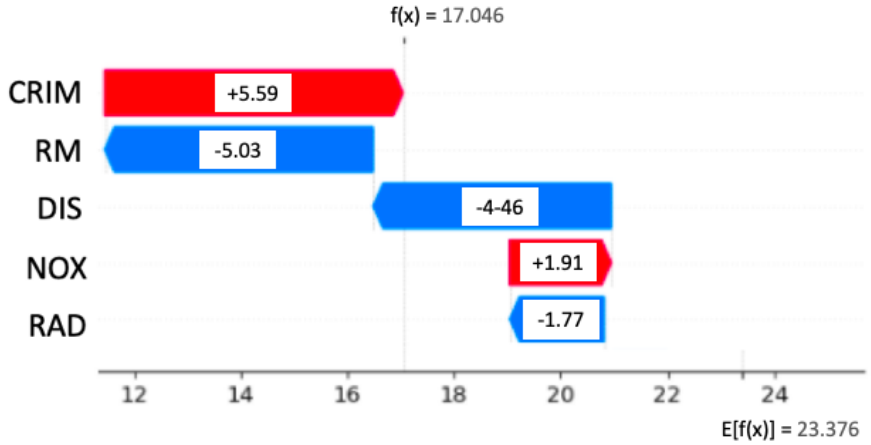


Figure 4.13 User Study: SHAP - Plot 1

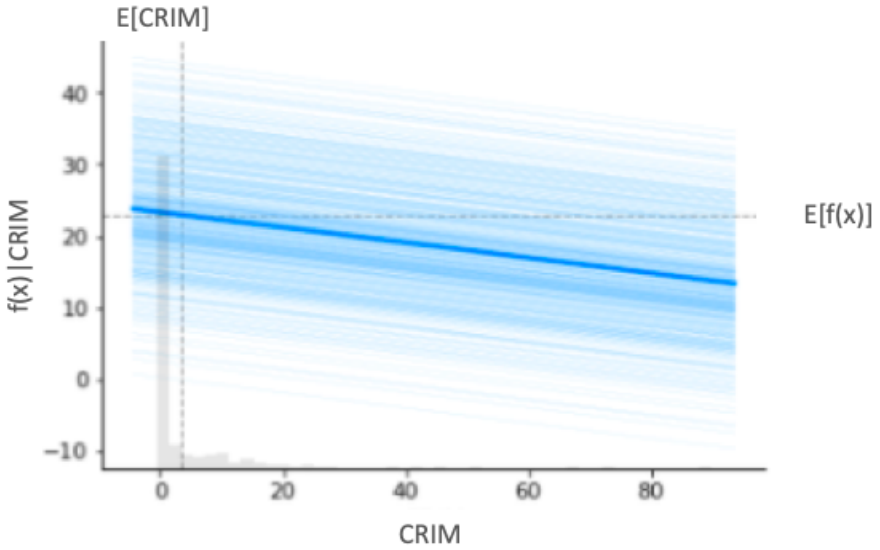


Figure 4.14 User Study: SHAP - Plot 2

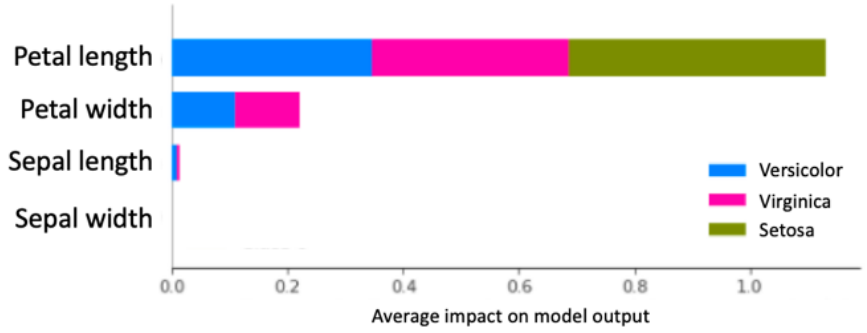


Figure 4.15 User Study: LIME - Plot 1

`ConvXAI`, given that the explanations used to construct the dialogues are not formed by `ConvXAI`.

Clarification Evaluation: As one contribution of `ConvXAI`, we have integrated clarification dialogues with explanation dialogues. This part evaluates how such clarifications are useful to different groups of participants.

Overall Evaluation: To evaluate the overall user experience, we combined the previous parts and asked participants to evaluate the overall performance of the dialogues considering both explanations (generated by external explainers) and clarifications (generated by `ConvXAI`).

Explanation Evaluation

To represent each group's evaluation, we calculated the mean score given by participants in each group for each question and explanation.

How different groups evaluate the overall explanations We computed Krippendorff's alpha coefficient, a statistical measure of the extent of agreement among users, with 0 indicating a complete absence of agreement and 1 indicating total agreement. The following alpha values were calculated for questions 1–5, respectively: 0.23, 0.61, 0.60, 0.70, 0.42. All but one of these values are less than 0.7 (the threshold considered by Krippendorff as acceptable for assessing a positive consensus [153]). Thus, we can conclude that different users perceive questions in different ways, with the biggest difference associated with the first question: *"I did not need support to understand the explanation"*.

How pairs of groups evaluate each explanation Extending the previous evaluation, we calculated the pairwise Krippendorff's alpha among groups. As listed in Table 4.3, the only alpha values greater than 0.7 are associated with questions 2 and 4. For question 2 (*I found that the explanation included all relevant known causal factors with sufficient precision and granularity*), the first (non-technical) and third (technical) groups provide similar evaluations; this indicates that, unlike the job function, the level of technology use is not a defining parameter in the perception of explanation usefulness. For question 4 (*I think that most people would learn to understand the explanation very quickly*), the result indicates that different groups, regardless of technology and experience levels, were sceptical about the possibility of learning the explanations. Such a result is aligned with our assumption

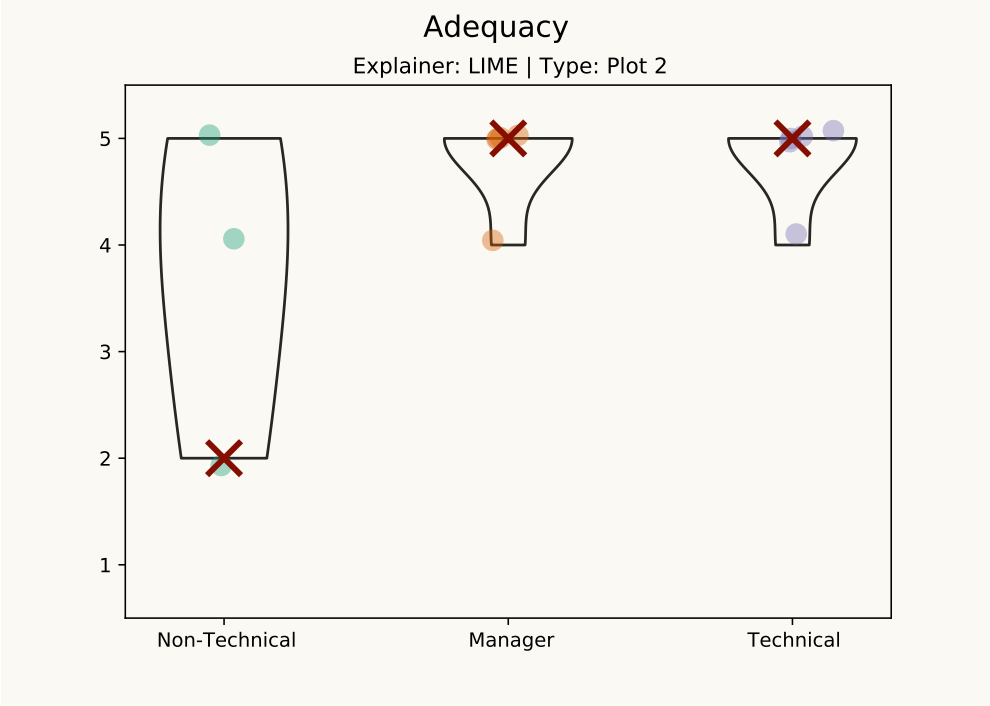


Figure 4.16 How groups evaluated Adequacy of explanations - The red X shows the mode of values

Table 4.3 Groups' pair-wise Krippendorff's alpha for each question

	Question 1			Question 2			Question 3			Question 4			Question 5		
	G1	**G2**	**G3**	**G1**	**G2**	**G3**	**G1**	**G2**	**G3**	**G1**	**G2**	**G3**	**G1**	**G2**	**G3**
G1															
G2	0.57			0.54			0.57			0.57			0.66		
G3	0	0.13		0.8	0.57		0.58	0.62		0.89	0.72		0.25	0.3	

that non-tailored explanations based on the user profile will lead to difficulties in users comprehending the explanation.

How different groups evaluate single explanations Focusing on how different groups perceive each explanation, we use violin plots to illustrate the intra-group agreement. Figures 4.16–4.20 show the responses given by various groups to a single explanation, i.e. a textual explanation generated for the iris dataset (the complete set of plots can be found in the Supplementary Materials). Note that we have added a random jitter of 0.04 to both the x and y axes to make the plots easier to read. Analysing these results, we find that group 2 (managers) has the highest variance of values, while group 3 (technical) demonstrates the lowest variance among groups. This indicates that, while groups 1 (non-technical)

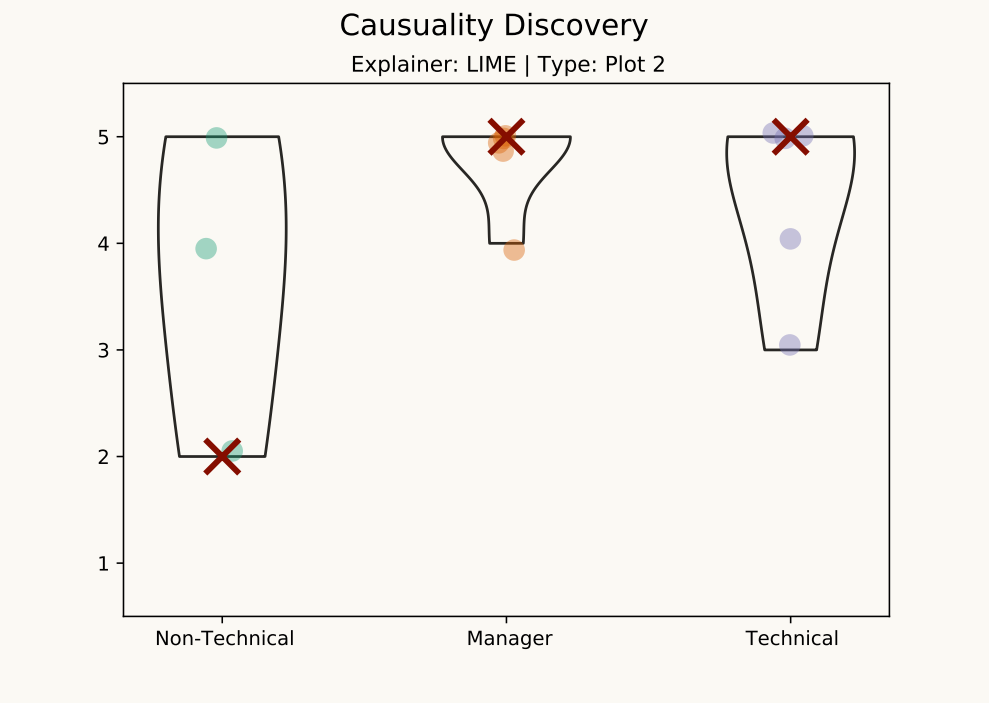


Figure 4.17 How groups evaluated the usefulness of explanations for discovering the causality of explanations - The red X shows the mode of values

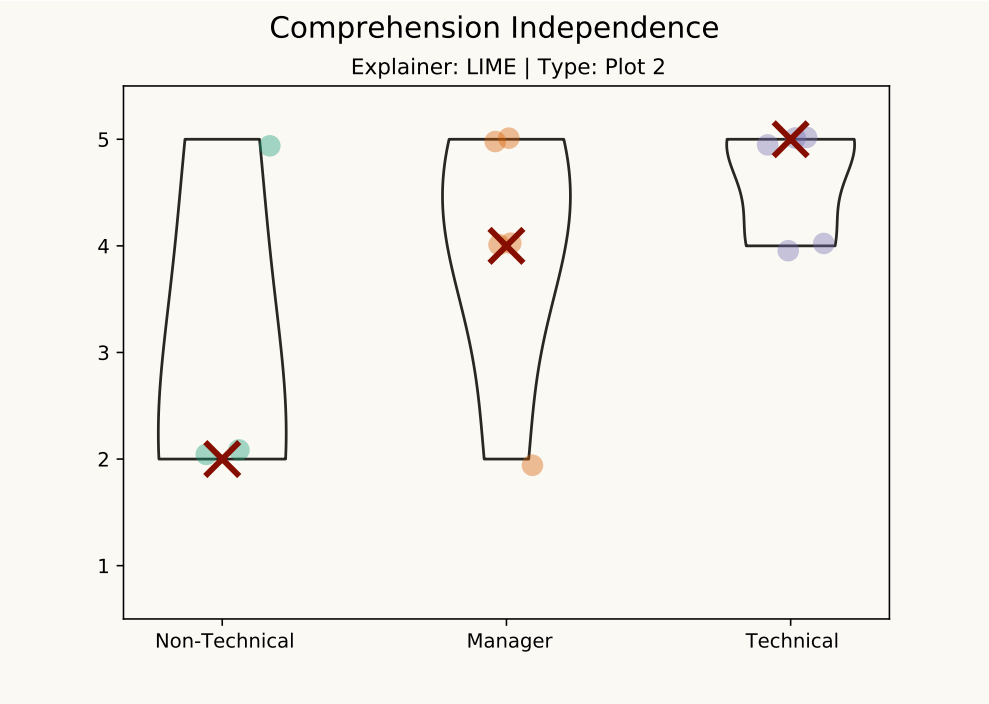


Figure 4.18 How groups evaluated the independence level in comprehending explanations - The red X shows the mode of values

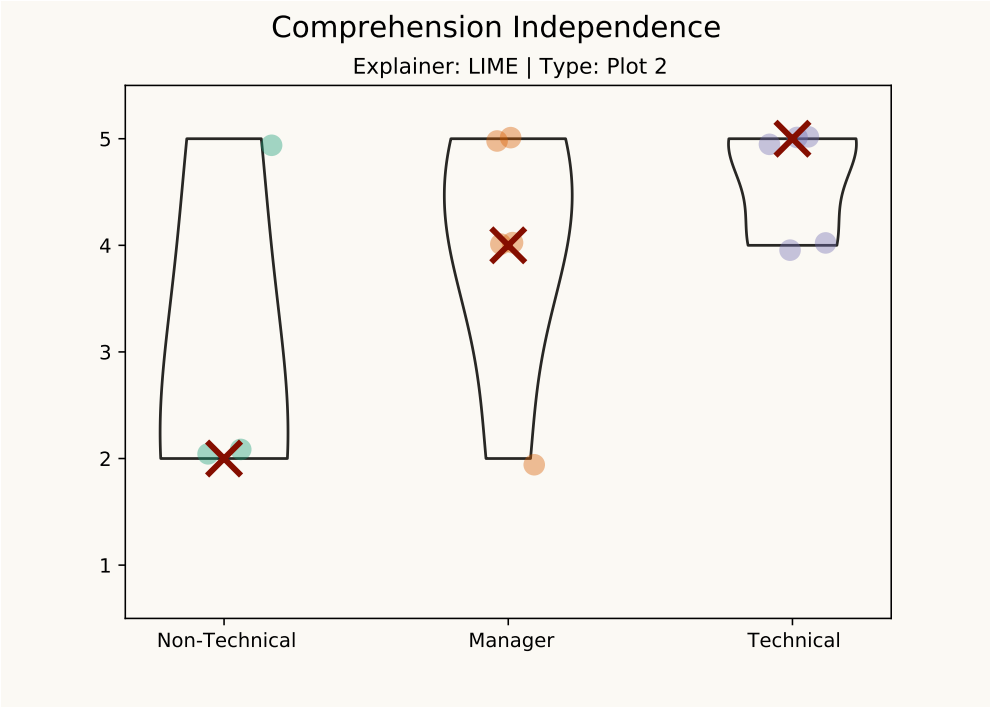


Figure 4.19 How groups evaluated the independence level in comprehending explanations - The red X shows the mode of values

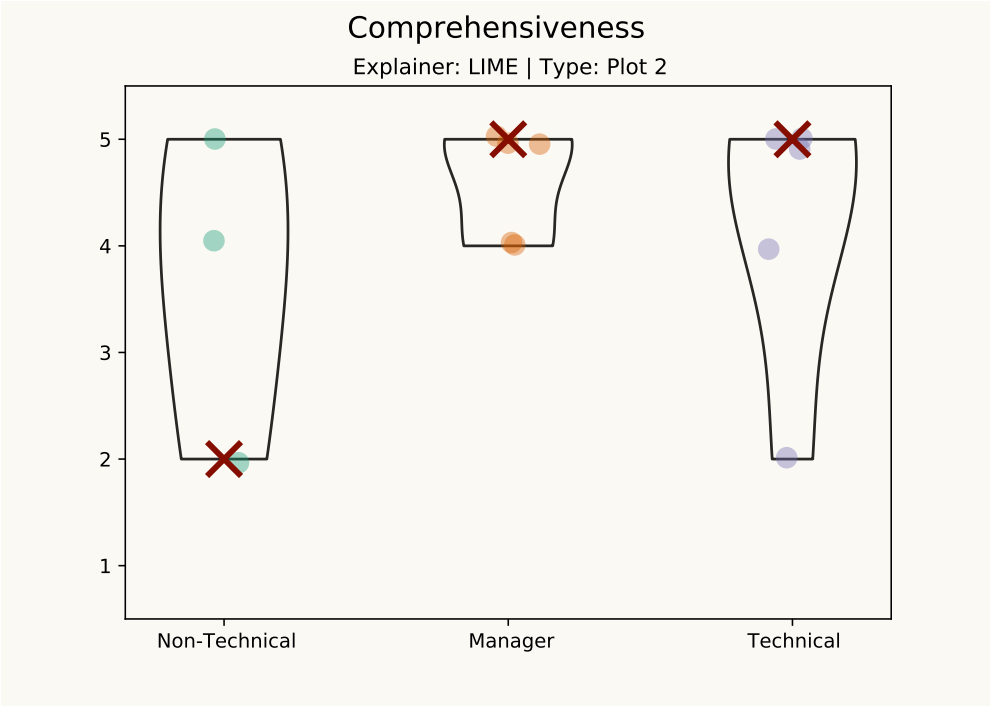


Figure 4.20 How groups evaluated how comprehensive are the explanations - The red X shows the mode of values

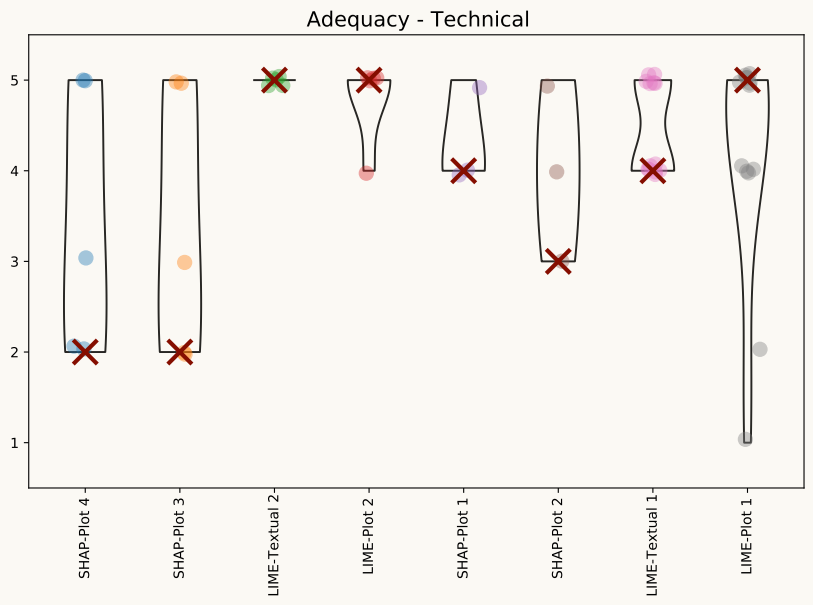


Figure 4.21 How technical group evaluated Adequacy of explanations - The red X shows the mode of values

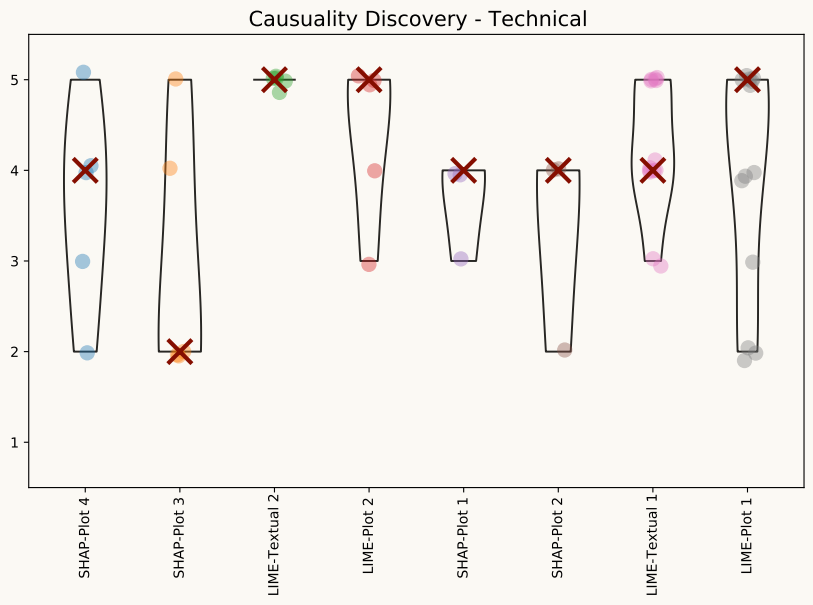


Figure 4.22 How technical group evaluated the usefulness of explanations for discovering the causality of explanations - The red X shows the mode of values

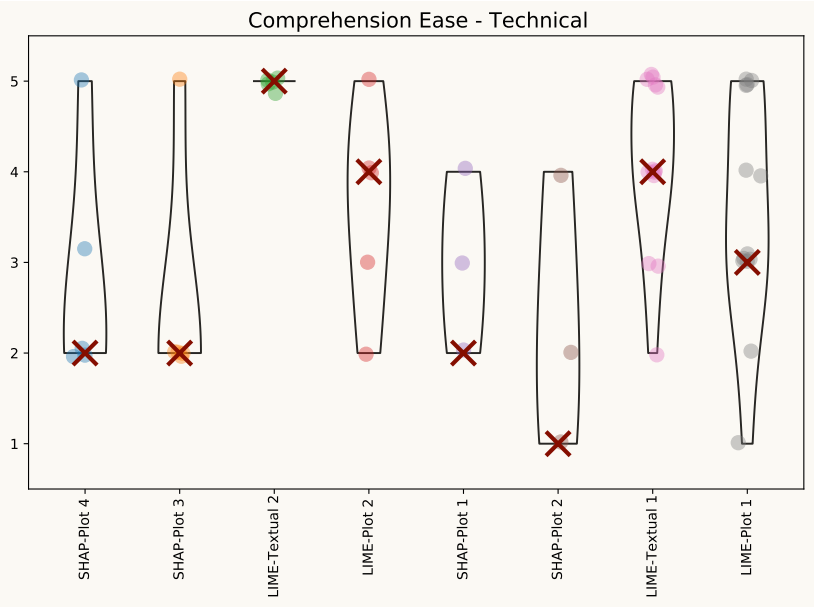


Figure 4.23 How technical group evaluated the comprehension ease of explanations - The red X shows the mode of values

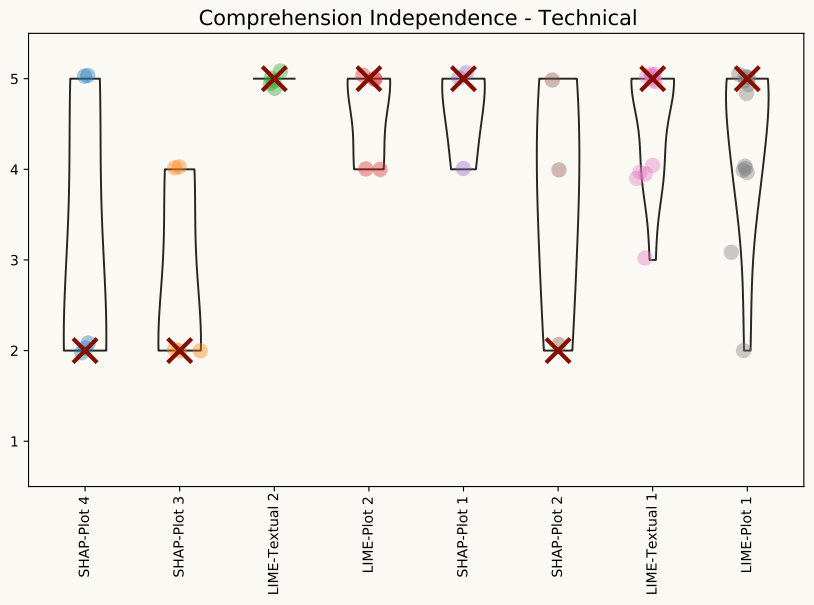


Figure 4.24 How technical group evaluated the independence level in comprehending explanations - The red X shows the mode of values

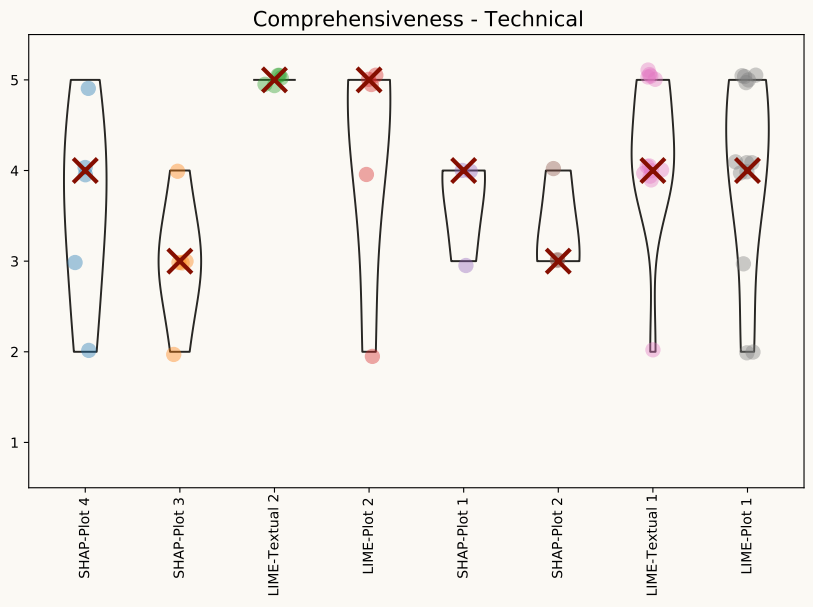


Figure 4.25 How technical group evaluated how comprehensive are the explanations - The red X shows the mode of values

and 3 (technical) can indicate a meaningful user profile concerning explanations, group 2 (managers) is too broad and should be further refined.

How users within each group evaluate overall explanations Using violin plots again, we analysed how the participants within each group perceived the overall explanations. Figures 4.21–4.25 show how the participants of group 3 (technical) evaluated each explanation through the five questions stated earlier in the text (the complete set of plots can be found in the Supplementary Materials). These plots show that textual explanations generally received the best scores, regardless of the participant group. In comparison, graphical explanations performed significantly worse for all five questions. Non-technical participants gave considerably lower scores to all questions than technical participants. However, in most cases, the scores from the non-technical group were similar to those of the managers. As expected, the various plots received different results among the three groups.

Clarification Evaluation

Figures 4.26–4.29 show the degree to which the participants found the provided clarifications useful for the selected explanations. Each participant showed their agreement with the phrase "The agent was able to understand the user's specific need and provide the required

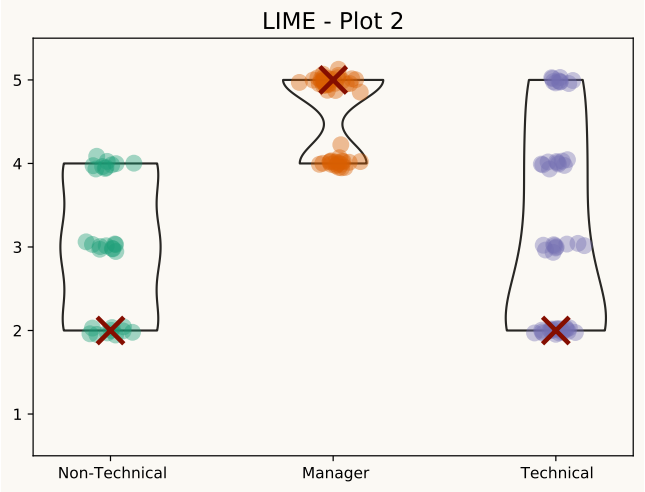


Figure 4.26 Clarifications scores given to Plot 2 generated by LIME explainer - The red X shows the mode of values

explanations and clarifications". Except for one clarification related to LIME - Plot 2 (see Figure 4.12), the mode value for all clarifications is at least 3, which corresponds to "Neither agree nor disagree". Surprisingly, the clarifications were most appreciated by the technical group, followed by the non-technical group. These results show that, except for a particular plot (see Figure 4.13), the managers had a similar opinion about the clarifications. At the same time, the technical and non-technical participants did not follow the same pattern.

Overall Evaluation

Figure 4.30 depicts the level of agreement with the phrase "The agent was able to understand the specific need of the user and provide the required explanations and clarifications" regarding each dialogue. In this case, the technical participants gave the highest scores (mode=5), closely followed by the other groups. Such high scores demonstrate that the dialogues generated by `ConvXAI` using external explainers can satisfy different groups of users divided by technology use and job function.

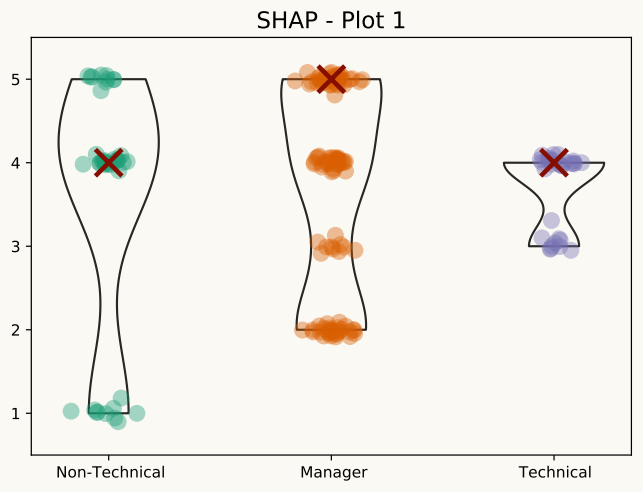


Figure 4.27 Clarifications scores given to Plot 1 generated by SHAP explainer - The red X shows the mode of values

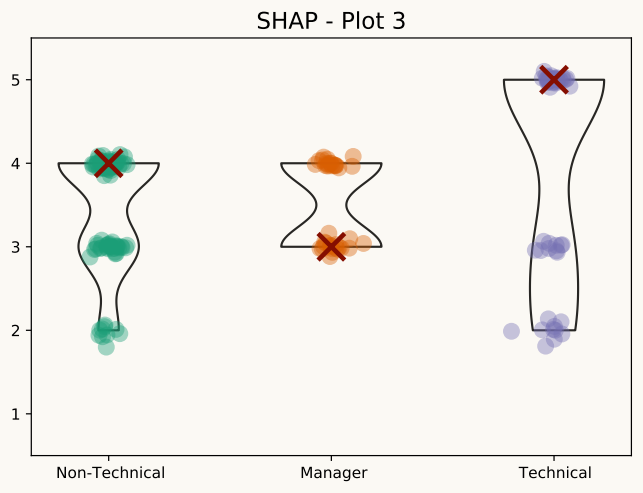


Figure 4.28 Clarifications scores given to Plot 4 generated by SHAP explainer - The red X shows the mode of values

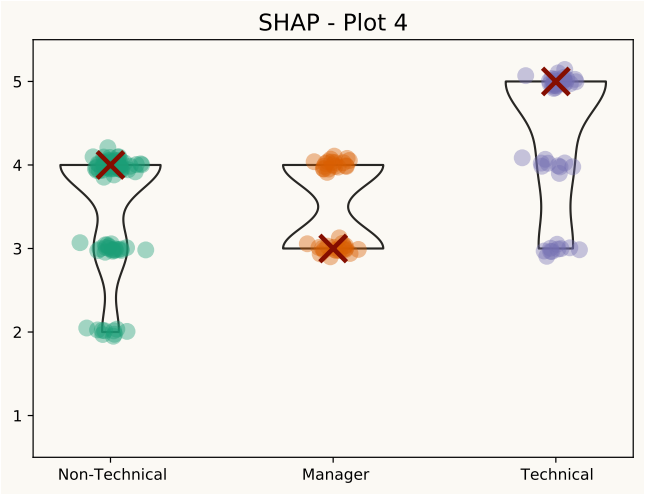


Figure 4.29 Clarifications scores given to Plot 4 generated by SHAP explainer - The red X shows the mode of values

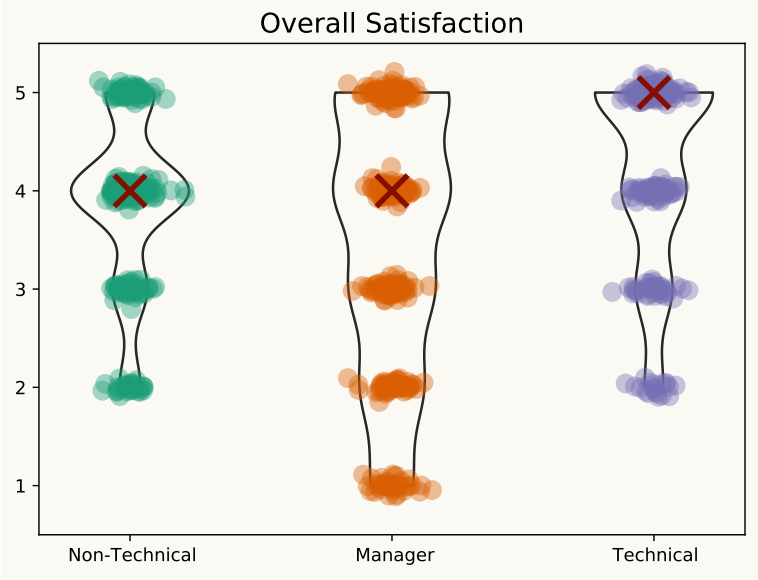


Figure 4.30 The overall scores given by groups to dialogues - The red X shows the mode of values

4.4 Conclusion

The ethical guidelines of the European Commission for trustworthy AI[6] state that transparency is one of the "Requirements of Trustworthy AI", in which:

- Explanations should be timely and adapted to the expertise of the stakeholder concerned;
- The AI system's capabilities and limitations should be communicated to AI practitioners or end-users in a manner appropriate to the use case at hand; and
- AI systems should consult stakeholders who may directly or indirectly be affected by the system throughout its life cycle.

Inspired by this, in this chapter we described a novel approach called `ConvXAI` for conveying explanations to end-users through interactive and multi-modal conversations, considering their level of knowledge and work experience. Such an approach is able to explain the inner workings of black box models and clarify the explanations through a dedicated module called the clarification dialogue.

To build `ConvXAI`, we formally extended the conversational explanation framework proposed by Madumal et al. [182] by introducing the clarification dialogue as a separate dialogue type. We further implemented our approach as an off-the-shelf Python tool.

Based on a user study including 45 participants, we were able to confirm that i) given the high Krippendorff alpha values (all greater than 0.7 except one), participants with different levels of technology use and work experience perceive explanations differently; ii) regardless of the dissimilarity between participants, they all prefer textual explanations over graphical ones; iii) regarding how different groups evaluate single explanations, group 2 (managers) exhibit the highest variance of values, while group 3 (technical) demonstrates the lowest variance among groups, and iv) `ConvXAI` is able to provide clarifications that increase the usefulness of the original explanations.

[6] https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai

Part II

HITL in LMI

5

Setting the stage on LMI

Taxonomies are a natural way to represent and organise concepts hierarchically. They are pivotal for machine understanding, natural language processing, and decision-making tasks. However, taxonomies are domain-dependent, usually have low coverage, and their manual creation and update are time-consuming and require domain-specific knowledge [87]. For these reasons, many researchers have tried to automatically infer semantic information from domain-specific text corpora to build or update taxonomies. Unlike the automated construction of new taxonomies from scratch, which is a well-established research area [291], the refinement of existing hierarchies is gaining in importance. Due to the evolution of human languages and the proliferation of online content, it is often required to improve existing taxonomies while maintaining their structure. To date, the most adopted approaches to enrich or extend standard *de-jure* taxonomies lean on expert panels and surveys; while these approaches entirely rely on human knowledge and have no support from the AI-side, a possible way through are word embeddings, as an outcome of the distributional semantics field. Word embeddings rely on the assumption that words occurring in the same context tend to have similar meanings. These methods are semi-supervised and knowledge-poor, thus suitable for large corpora and evolving scenarios.

In recent years, lexical taxonomies and distributional semantics have gained momentum in NLP applications. Lexical taxonomies are a natural way to organise human knowledge in a hierarchical form and provide a formal description of concepts and their relations, supporting syntactic and semantic exchanges. Contextually, word embeddings have gained remarkable popularity in computational linguistics. They are real-valued word representations, able to extract linguistics patterns and lexical semantics from large corpora. However, despite their wide usage, finding a unified measure accounting for both knowledge-based and distributional resources is still an open problem.

Moreover, despite the extensive usage of word embeddings in a wide variety of modern Natural Language Processing [269], Understanding [108] and Generation [320] tasks, the evaluation of the performance of such representations is still an open debate. There is not a unique definition of what either an "effective" or a "performant" assessment measure is (see, e.g. [21, 224]); researchers tend to agree that the optimal vector representation depends on the use it will serve [247]. One thing is clear: given the high sensitivity of word embeddings to small changes in hyperparameters during their generation [166, 51], it is crucial to have a reliable measure of evaluation that can produce a performant semantic representation based on the intended scope and the information they have to embed.

In our case, selecting a word vector model that represents and preserves taxonomic similarity relations would allow us to generate a unified representation of knowledge-based and data-driven lexical features and, at the same time, would enable several NLP applications. Some of them are related to the maintenance and update of the taxonomy itself, like taxonomy refinement, enrichment, and learning. Others are downstream tasks that rely on underlying structured knowledge representation, like personalised recommendations for online retailers [319], query understanding for search engines [130], and text understanding [302], to cite a few of them.

Semantic similarity represents a particular case of semantic relatedness [237], considering only the co-hyponymy and synonymy relations. For instance the words *cat* and *tiger* are more similar than the words *jungle* and *tiger*, while the latter pair seems to be more related. The semantic similarity strongly depends on the context. For this reason, it is useful to find an automated way to compute the similarity between words in a domain-dependent taxonomy. The majority of current semantic similarity metrics, they have two main drawbacks: first,

when a word has multiple senses, those methods compute a value of similarity for each word sense and then consider only the highest; second, they consider the structure of the taxonomy, thus the relationship between taxonomic concepts, none of those measures accounts for the number of words belonging to those concepts.

Evaluating the intrinsic quality of vector space models, as well as their impact when used as input for specific tasks (*aka*, extrinsic quality), has a very practical significance (see, e.g. [45]), as this affects the trustworthiness of the overall process or system in which they are used (see, e.g., [293, 200]). We may argue that the well-known principle *"garbage-in, garbage-out"* of the data quality research field also applies to word embedding, as *the lower the quality of the embeddings, the lower the effectiveness of the tasks based on them.*

This thought, along with the need to keep updated with current taxonomies, inspired part II of this thesis, which is framed within the research activities of an ongoing European tender[1] for the Cedefop EU Agency[2]. The project aims at realising a European system to collect and classify Online Job Vacancies (OJVs)[3] for the whole EU country members through machine learning [36]. OJVs are encoded within ESCO [4], an extensive taxonomy with 2,942 occupations and 13,485 skills serving as a *lingua franca* to compare labour market dynamics across borders.

5.1 The Significance of Analysing Job Ads

In recent years, the European Labour demand conveyed through specialised Web portals and services has grown exponentially. This also contributed to introducing the term *"Labour Market Intelligence"* (LMI), which refers to the use and design of AI algorithms and frameworks to analyse Labour Market Data for supporting decision making (see, e.g., [317, 285, 98]).

Nowadays, the problem of monitoring, analysing, and understanding labour market changes (i) timely and (ii) at a fine-grained geographical level has become practically

[1] Real-time Labour Market information on Skill Requirements: Setting up the EU system for online vacancy analysis. https://goo.gl/5FZS3E (2016)

[2] The European Center for the Development of Vocational Training https://www.cedefop.europa.eu/

[3] An Online Job Vacancy (OJV, *aka*, job offers) is a document containing a *title* - that shortly summarises the job position - and a *full description*, that advertises the skills a candidate should hold.

[4] The European Taxonomy of Occupations, Skills & Qualifications https://ec.europa.eu/

significant in our daily lives. Recently, machine learning has been applied to compute the effect of robotisation within occupations in the US labour market [86] as well as to analyse skill relevance in the US standard taxonomy O*Net [7], just to cite a few. In 2016 the EU Cedefop agency - aimed at supporting the development of European Vocational Education and Training - launched a European tender for realising a machine-learning-based system able to collect and classify Web job vacancies from all 28 EU country members using the ESCO hierarchy for reasoning over the 32 languages of the Union [1]. Preliminary results of this project have focused on analysing lexicon extracted from OJVs as well as to identify novel occupations and skills (see, e.g. [96, 93, 36, 35, 186]).

As one might note, the use of classified OJVs and skills, in turn, enables several third-party research studies to understand and explain complex labour market phenomena. Just to give a very recent few examples, in [64] authors used OJVs for estimating the impact of AI in job automation and measuring the impact of digital/soft skills within occupations; In May 2020, the EU Cedefop Agency started using those OJVs to build an index named Cov19R that identifies workers with a higher risk of COVID-19 exposure, who need greater social distancing, affecting their current and future job performance capacity[5]. While on the one side, ESCO allows comparing different labour markets, on the other side, it does not encode the characteristics and peculiarities of such labour markets in terms of skills advertised - which vary Country by Country - and the meaning of terms that are used differently based on the level of maturity of local labour markets.

5.2 Preliminaries and Related Work

In this section, we survey state of the art related to word embeddings and their evaluation methods and approaches to taxonomy learning and refinement.

[5]https://www.cedefop.europa.eu/en/news-and-press/news/cedefop-creates-cov19r-social-distancing-risk-index-which-eu-jobs-are-more-risk

5.2.1 Word Embeddings

Evaluating the intrinsic quality of vector space models, as well as their impact when used as the input of specific tasks (*aka*, extrinsic quality), has a very practical significance (see, e.g. [274, 45]), as this affects the believability[6] of the overall process or system in which they are used. In essence, we may argue that the well-known principle *"garbage-in, garbage-out"* that characterises the data quality research in many domains (see, e.g. [200, 34, 33, 199]) also applies to word embedding, that is, *the lower the quality of the word embeddings, the lower the performance of the tasks that are based on them*.

Word Embeddings are vector representations of words based on the hypothesis that words occurring in a similar context tend to have a similar meaning. Words are represented by semantic vectors, which are usually derived from a large corpus using statistical language models and co-occurrence statistics, and their use improves learning algorithms in many NLP tasks. Two powerful methods to induce word embeddings are neural networks training [63, 201] and co-occurrence matrix factorisation [223, 165].

These techniques consider each word as a distinct vector and ignore the morphological similarity among them. More recently [29] developed a version of the continuous skip-gram model [202] which considers subword information. This architecture, called fastText, allows capturing morphological similarity among words (e.g. typos, singular-plural, words with the same root, etc.). In essence, the main difference between fastText and word2vec is what they consider as the atomic entity of the corpus. In the case of word2vec (and also GloVe [223]) this entity is the word, while fastText considers character *n*-grams as atomic entities.

While the mentioned methods basically use a single corpus in the generation phase of word representations, in the past decade, several authors attempted to create directed or nudged embeddings by associating external semantic lexicons for manipulating similarities and relatedness in the embeddings. These manipulations can happen both during (joint learning e.g. [145]) and after (retrofitting e.g. [82]) the generation process. Similarly, [109] associates a taxonomy (WordNet) with word embeddings using pairs of related words to constrain the learning process. This work has two main differences from ours: (i) Unlike their work, the

[6]Here the term believability is intended as "the extent to which data are accepted or regarded as true, real and credible"[293]

method we propose in Chapter 6, called `MEET` not only considers the similarities extracted from the taxonomy but also the hierarchical structure (ii) In their work, the interaction between taxonomy and the embeddings happens directly in the generation phase while in `MEET` the taxonomy is not engaged in the generation phase as it is used to evaluate the embeddings. Such a method offers a greater level of flexibility since not all the embedding generation methods can be easily bonded with external sources in their generation phase.

Formally, given a word w, and a dictionary of size G, G_{w} is the set of n-grams of size G appearing in w. Denoted as z_g vector representation of the n-gram g, w will be represented as the sum of the vector representation of its n-grams and the score associated to the word w as:

$$\int(\mathrm{w},\mathrm{c}) = \sum_{g \in G_{\mathrm{w}}} z_g^{\intercal} v_{\mathrm{c}} \tag{5.1}$$

where v_{c} is the vector representing the context. This simple representation allows one to share information between words, and this makes it useful to represent rare words, typos, and words with the same root.

Other embedding models have been evaluated along with fastText. Nevertheless, none of them fit our conditions. Neither classical embedding models [202, 223] nor embeddings specifically designed to fit taxonomic data [82, 145] consider subword information. Moreover, they cannot be easily bonded with external sources in their generation phase, and this would reduce the flexibility of `TaxoVec`. Regarding hyperbolic and spherical embeddings like HyperVec [210] or JoSe [198], we discarded them since (i) they also don't consider subword information, which is important for short text and many words with the same root (e.g. engineer-engineering,developer-developing) like OJVs, and (ii) HyperVec uses hypernym-hyponym relationships for training, while we train our models on a text corpus which has not such relationships. Finally, we considered context embeddings (see e.g. [73]). However, contextual embeddings represent words based on their context, thus capturing the uses of words across varied contexts. Thus is not suitable for our case, where we aim to compare words in a corpus and their similarity with words of taxonomy, with a given sense.

Evaluation Methods for word embeddings When it comes to classifying embedding evaluation methods, researchers almost univocally agree on *intrinsic* vs *extrinsic* division (see, e.g., [247, 290]).

Intrinsic metrics attempt to increase the syntactic or semantic relationships between words either by directly getting human judgments (Comparative intrinsic evaluation [247]) or by comparing the aggregated results with the pre-constructed datasets (Absolute intrinsic evaluation). The comparison of word vectors and human assessment is made in various ways; here we mention two of the most important ones: *Semantic Similarity* which compares the similarity between word vectors and their similarity ranks given by assessors and *Analogical Reasoning* (see, e.g. [209]).
Some of the limitations of *intrinsic metrics* are: (i) suffering from word sense ambiguity(faced by a human tester) and subjectivity (see, e.g. [175, 290]); (ii) facing difficulties in finding significant differences between models due to the small size and low inter-annotator agreement of existing datasets and (iii) need for constructing judgement datasets for each language [150].

Extrinsic metrics, on the other hand, perform the evaluation by using embeddings as features for specific downstream tasks. For instance, question deduplication [159], Part-of-Speech (POS) Tagging, Language Modelling [228] and Named Entity Recognition (NER) [91], just to cite a few. As a drawback, extrinsic metrics (i) are computationally heavy [290]; (ii) have high complexity of creating gold standard datasets for downstream tasks, and (iii) lack performance consistency on downstream tasks [247].

The metrics we propose in this chapter are similar to the *intrinsic thesaurus-based evaluation metrics* as they use an expert-constructed taxonomy to evaluate the word vectors as an intrinsic metric [22, 273]. At the same time, typically, these kinds of approaches evaluate embeddings by their correlation with manually crafted lexical resources, like expert rating of similarity or relatedness between hierarchical elements. However, those resources are usually limited and hard to create and maintain. Furthermore, human similarity judgments evaluate ex novo the semantic similarity between taxonomic, but the structure of the taxonomy already encodes information about the relations between its elements. In this research, instead, we exploit the information encoded in an existing taxonomy to build a benchmark for the evaluation of word embeddings. As far as we know, this is the first attempt to encode

similarity relationships from a manually built semantic hierarchy into a distributed word representation automatically extracted from a text corpus.

5.2.2 Taxonomies

A definition of Taxonomy To better describe `TaxoVec`(See Chapter 6), in the following part, we introduce and formalise the notion of semantic hierarchy. Based on [184], the building blocks of a semantic hierarchy for a specific domain S can be defined as follows:

- A set $\mathscr{C}$ of concepts (nodes) c belonging to the domain D;
- A set W of possible words (or entities) belonging to the domain D. Each word w can be assigned to none, one or multiple concepts $c \in \mathscr{C}$;
- A taxonomic relation (edge) $H^c \subseteq \mathscr{C} \times \mathscr{C}$, which is a directed relation between elements in $\mathscr{C}$, i.e., where $H^c(c_1, c_2)$ means that c_1 is a subconcept of c_2. The relation $H^c(c_1, c_2)$ is also called $IS-A$ relation (c_1 $IS-A$ subconcept of c_2).

The direction of H^c is from the most specific concept to the most generic. As a consequence, the concepts with the finest granularity have an in-degree of 0, i.e. they do not have any incoming edges.

Taxonomy-based Semantic Similarity

In past literature, several researchers attempted to measure semantic similarity between words in a taxonomy, expressed as a similarity between the concepts to which the two words belong. Those measures can be roughly divided into two main categories: those that exploit the path connecting two concepts and those that are based on the Information Content (IC) of the concepts. The most important measures belonging to these two categories, as assessed by different researchers [158, 15], are the following.

Path-based measures These measures employ the path connecting two concepts to estimate their similarity.

The simplest approach is to use the *shortest path* to assign a similarity score:

$$sim_{sp}(c_1, c_2) = \frac{1}{\phi(c_1, c_2) + 1}$$

where $\phi(c_1, c_2)$ is the shortest path between c_1 and c_2.

Leacock and Chodorow [160] scale the path similarity by the depth of the taxonomy:

$$sim_{lc}(c_1, c_2) = -\log \frac{\phi(c_1, c_2)}{2 \times max\_depth}$$

Wu and Palmer [304] consider the position of $LCA(c_1, c_2)$, the Lowest Common Ancestor of c_1 and c_2:

$$sim_{wup}(c_1, c_2) = \frac{2 \times \phi(r, LCA)}{\phi(c_1, LCA) + \phi(c_2, LCA) + 2 \times \phi(r, LCA)}$$

where r is the root node. Note that, to simplify the notation, we refer to $LCA(c_1, c_2)$ simply as LCA. For this class of methods in case of polysemy, i.e. words belonging to several concepts, the minimum $\phi(c_1, c_2)$ is considered, thus the maximum similarity.

Information content-based measures The IC-based approach was introduced by Resnik [237]. According to information theory, the IC (or self-information) of a concept $c \in \mathscr{C}$ can be approximated by its negative log-likelihood:

$$IC(c) = -\log p(c)$$

Where $p(c)$ is the probability of encountering the concept c.

In the case of a taxonomy, $p(c)$ is monotonic and increasing with the rank of the taxonomy: if c_1 $IS-A$ c_2 then $p(c_1) \leq p(c_2)$. Moreover, the probability of the root node is 1. Concepts probabilities are computed simply as relative frequencies in a text corpus:

$$\hat{p}(c) = \frac{freq(c)}{N} = \frac{\sum_{n \in words(c)} count(n)}{N}$$

Where $words(c)$ is the set of words subsumed by concept c and N the total number of occurrences of words in the corpus that are also present in the taxonomy. Therefore *Resnik* defines the similarity between two concepts c_1 and c_2 as:

$$sim_{res}(c_1, c_2) = IC(LCA)$$

and the similarity between two words as:

$$sim_{res}(w_1, w_2) = \max_{\substack{c_1 \in s(w_1), \\ c_2 \in s(w_2)}} IC(LCA)$$

Where $s(w_1)$ and $s(w_2)$ are all the possible senses of w_1 and w_2 respectively.

Jiang-Conrath [134] built a measure of similarity using the self-information of c_1 and c_2 as well:

$$sim_{jcn}(c_1,c_2) = \frac{1}{IC_{res}(c_1)+IC_{res}(c_2)-2\times IC_{res}(LCA)}$$

And a similar measure, is built by Lin [170]:

$$sim_{lin}(c_1,c_2) = \frac{2\times IC_{res}(LCA)}{IC_{res}(c_1)+IC_{res}(c_2)}$$

Similarly to Resnik, these last two methods consider the two concepts with the highest Resnik similarity in the case of multiple word senses.

Other IC-based measures, sometimes called *intrinsic IC-measures* use the structure of the taxonomy, instead of an external corpus frequency, to compute $\hat{p}(c)$. For instance Seco et al. [251] compute the IC(c) as:

$$IC_{seco}(c) = 1 - \frac{\log(|descendants(c)|+1)}{\log N_c}$$

where N_c is the number of concepts in the taxonomy and $|descendants(c)|$ the number of subconcepts of c.

They have two main drawbacks: first, when a word has multiple senses, those methods compute a value of similarity for each word sense and then consider only the highest; second, they consider the structure of the taxonomy, i.e. the relationship between taxonomic concepts, none of those measures accounts for the number of words belonging to those concepts.

In Sec. 6.1.1 we present the `HSS`, a measure for semantic similarity considering both multiple word senses and the cardinality of taxonomic concepts in terms of entities (words) that they subsume. This measures, that we developed in [185], has proven to be useful in several applications, like taxonomy enrichment [92, 96], refinement [187] and job-skill mismatch analysis in the field of labour market [97].

Taxonomy Learning

As the backbone of ontologies, taxonomies, also called semantic hierarchies [184], have received much attention from many researchers. In the past years, hierarchies based on manually-crafted semantic resources have formed the knowledge base for several important applications in the semantic web, including many NLP tasks. However, they include only a

limited number of concepts and relationships. Besides, supplementing and maintaining a manually built hierarchy are cumbersome and time-consuming tasks, which often result in a bottleneck in the knowledge acquisition process [183]. For this reason, a wide range of techniques has been developed to enrich and revise semantic hierarchies automatically.

Recently, the introduction of Distributional Semantic Models (DSM) has boosted approaches to learning semantic hierarchies. For example, [87] proposes a method for the construction of semantic hierarchies based on word embeddings. In [280], authors propose a model that learns a high-dimensional embedding for the existing and new concepts of the taxonomy. In [26], authors combine distributional semantic representations induced from text corpora with manually constructed lexical-semantic networks.

Though all these approaches are very relevant, none of them neither employs nor exploits in any way the structure of an existing taxonomy to represent the taxonomy itself through word embeddings.

Taxonomy Refinement

While the automatic extraction of taxonomies from text corpora has received considerable attention in past literature [291], the evaluation and refinement of existing taxonomies are growing in importance in all the scenarios where the user wants to update a taxonomy while maintaining its structure rather than rebuilding it from scratch. Most of the existing methods are either domain-dependent [212] or related to lexical structures specific to the existing hierarchies [225]. Two recent TExEval tasks at SemEval-2016 [30, 31] introduce a setting for the evaluation of taxonomies extracted from a test corpus, using standard precision, recall and F1 measures over gold benchmark relations from WordNet and other well-known resources, resorting to human evaluation for relations not included in the benchmark. Though interesting, this methodology relies on existing resources, which to some extent, could be inaccurate. [12] employ Poincaré embeddings to find a child node that is assigned to the wrong parent. Although the goal of this research is quite similar to ours, in the detection of outliers, they use Poincaré embeddings trained on the taxonomy without considering any external corpus, thus information on the use of taxonomic terms. Though all these approaches are very relevant, none of them exploits in any way the structure of an existing hierarchy to preserve taxonomic relations in word embeddings.

6

Embedding Evaluation Through Semantic Similarity

Lexical taxonomies and distributional representations are largely used to support a wide range of NLP applications, including semantic similarity measurements. Recently, several scholars have proposed new approaches to combine those resources into a unified representation preserving distributional and knowledge-based lexical features. Regardless of the crucial role embeddings play in NLP and particularly in LMI, they often carry shortcomings that, if not addressed adequately, may negatively impact the AI system. Powell et al. [227] argument that these issues can manifest themselves as:

- Capturing uneven semantic and syntactic representations
- Various forms of bis
- Diffuse representations as a result of the existence of multiple senses
- Poor contextual representation due to scarcity of data
- skewness toward particular fields/subjects

While many works propose quality evaluation methods for embedding, the majority are cumbersome and labour-intensive or highly rely on algorithmic solutions, which do not

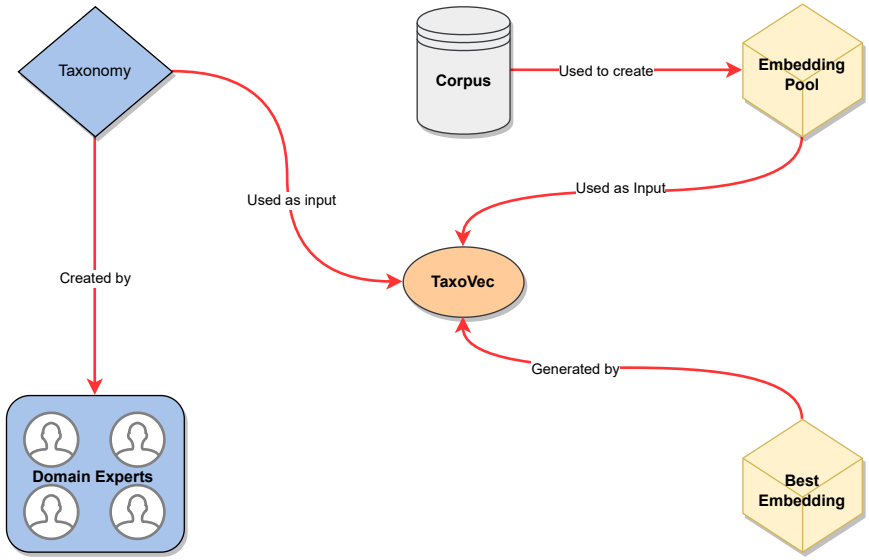


Figure 6.1 A general overview of the integration of human-in-the-loop approach in Embedding evaluation task

guarantee to catch all the peculiarities of human language.

Another issue regarding the quality of embeddings, especially in the LMI field, is that often AI-based LMI systems are based on a single or multiple predefined taxonomies like ESCO[1], SOC[2] and O*NET[3] with each having a different hierarchy, the number of levels, groups and sub-groups and concepts. If not considered in embedding generation and evaluation, such differences can often cause significant problems in downstream tasks such as occupation classification and skill extraction.

In this chapter, using HITL paradigm, we propose and implement `TaxoVec`, a novel approach that, given a pool of embeddings, based on their ability to preserve taxonomic similarity, selects the top embeddings. It further chooses the best embedding based on how they represent the opinion of a group of LMI experts. Figure 6.1 depicts this process which is later detailed throughout the chapter.

[1]https://esco.ec.europa.eu/en
[2]https://www.bls.gov/soc/2018/home.htm
[3]https://www.onetcenter.org/taxonomy.html

6.1 Methodology

In this section, we present `TaxoVec`, which is composed of three steps. In *Step 1* we define a measure of similarity within a semantic hierarchy, which will serve as a basis for the embeddings evaluation. In *Step 2* we create a tool for computing Semantic Similarity using our metric, the `HSS`, and the other state-of-the-art measures. In *Step 3* we generate embeddings from a large text corpus, and we identify which one better preserves the taxonomic relations of semantic similarity; then we utilise the embeddings to perform four NLP tasks, namely *Categorisation*, *Sentiment classification*, *Hypernym detection*, and *Synonym detection*.

6.1.1 Step 1: Hierarchical Semantic Similarity (`HSS`)

Following, we present and evaluate the measure of semantic similarity in taxonomies that we introduced in [185], the Hierarchical Semantic Similarity (`HSS`), that will be used for the evaluation of the embeddings in Section 6.1.3. Similarly to [251] we compute $\hat{p}(c)$ using an intrinsic measure, exploiting the structure of the taxonomy instead of an external corpus. However, the `HSS`, differently from [251], which uses only the number of taxonomic concepts, considers also the entities of the taxonomy:

$$\hat{p}(c) = \frac{N_c}{N} \tag{6.1}$$

where N is the cardinality, i.e., the number of entities (words) of the taxonomy and N_c the sum of the cardinality of the concept c with the cardinality of all its hyponyms. Note that $\hat{p}(c)$ is monotonic and increases with granularity, thus respects our definition of p (see 5.2.2).

Now, given two words w_1 and w_2, Resnik defines $c_1 \in s(w_1)$ and $c_2 \in s(w_2)$ to be all the concepts containing w_1 and w_2 respectively, i.e. the *senses* of w_1 and w_2. Therefore, there are $|s(w_1)| \times |s(w_2)|$ possible combinations of their word senses, where $|s(w_1)|$ and $|s(w_2)|$ are the cardinality of $s(w_1)$ and $s(w_2)$ respectively. We can now define $\mathscr{L}$ as the set of all the lowest common ancestor for all the combinations of $c_1 \in s(w_1), c_2 \in s(w_2)$.

The hierarchical semantic similarity between the words w_1 and w_2 can therefore be defined as:

$$sim_{\texttt{HSS}}(w_1, w_2) = \sum_{\ell \in \mathscr{L}} \hat{p}\,(\ell = LCA \mid w_1, w_2) \times IC(LCA) \tag{6.2}$$

Where $\hat{p}\,(\ell = LCA \mid w_1, w_2)$ is the probability of *LCA* being the lowest common ancestor of w_1, w_2, and can be computed as follows applying the Bayes theorem:

$$\hat{p}\,(\ell = LCA \mid w_1, w_2) = \frac{\hat{p}\,(w_1, w_2 \mid \ell = LCA)\,\hat{p}\,(LCA)}{\hat{p}\,(w_1, w_2)} \tag{6.3}$$

We define N_ℓ as the cardinality of ℓ and all its descendants.

Now we can rewrite the numerator of Eq. 6.3 as:

$$\hat{p}\,(w_1, w_2 \mid \ell = LCA)\,\hat{p}\,(\ell = LCA) = \frac{S_{<w_1, w_2> \in \ell}}{|descendants(\ell)|^2} \times \frac{N_\ell}{N}. \tag{6.4}$$

Where the first leg of the *rhs* is the class conditional probability of the pair $< w_1, w_2 >$ having ℓ as the lowest common ancestor and the second one is the marginal probability of the class ℓ. The term $|descendants(\ell)|$ represents the number of subconcepts of ℓ. Since we could have at most one word sense w_i for each concept c, $|descendants(\ell)|^2$ represents the maximum number of combinations of word senses $< w_1, w_2 >$ which have ℓ as lowest common ancestor. $S_{<w_1, w_2> \in LCA}$ is the number of pairs of senses of word w_1 and w_2 which have *LCA* as lower common ancestor.

The denominator of Eq. 6.3 can be written accordingly as:

$$\hat{p}\,(w_1, w_2) = \sum_{k \in \mathscr{L}} \frac{S_{<w_1, w_2> \in k}}{|descendants(k)|^2} \tag{6.5}$$

Evaluation of the `HSS` This step aims to compare the `HSS` with the previously presented measures of semantic: WUP, LC, shortest path, Resnik, Jiang-Conrath, Lin, and Seco. The evaluation is composed by two tasks: *semantic similarity* and *word clustering* (also called *concept categorisation*). In the first one, we measure how the different measures of semantic similarity are correlated with a gold benchmark. In the second one, we measure how words belonging to the same semantic group in a gold benchmark are similar to each other. To do this, following the state of the art [15, 8], we adopt a set of human-annotated resources, considered the gold benchmark for semantic similarity and word categorisation.

6.1.2 Step 2: A Tool for Computing Semantic Similarity

Semantic similarity can be useful for a wide number of tasks. The embedding selection that we perform in Section 6.1.3 is only one of them. However, as highlighted in the introduction, hand-crafted semantic similarity resources are usually not updated and are limited in coverage. For this reason, we decided to implement the `HSS` and all the other automatic semantic similarity measures considered in Section 5.2.2 as a fully-fledged python package. This is going to facilitate the user that wants to use it. This library, called *TaxoSS* (Taxonomic Semantic Similarity), allows computing taxonomic-based similarity (using WordNet 3.0) as well as corpus-based similarity measures. For the latter, there is a default measure based on the English Wikipedia dump of the year 2008, but the user can also use a different corpus. All the implementation details are reported in Section 6.2.2.

6.1.3 Step 3: Embeddings Selection and Evaluation of the Best Embedding

In this step, we generate a variety of vector representations of a large text corpus through FastText and, by an intrinsic evaluation, we select the one that better represents the taxonomy, that is we want the similarity between word vectors to reflect as much as possible the semantic similarity between words in the taxonomy. Then we perform an extrinsic evaluation to compare the embedding model selected through `HSS` with the models selected by other semantic similarity measures.

Embeddings Selection

Following the previous literature [22, 247, 119], which correlates the cosine similarity between pairs of word vectors with human scores of relatedness/similarity, we assess the goodness of a vector model by the Pearson correlation coefficient between the cosine similarity of pairs of word vectors and their taxonomic semantic similarity. The taxonomic semantic similarity can be measured with all the metrics presented before or by experts, like in the case of human-crafted resources.

This kind of evaluation was developed in [22] with the name of *Semantic Relatedness*, where the authors use MEN as a measure of similarity. In [119] the authors present a new

dataset, SimLex-999, which explicitly measures similarity rather than relatedness, to foster the development of models that reflect word similarity instead of relatedness.

In this step, we select the best embeddings that maximise the correlation between cosine similarity and semantic similarity as it is computed by HSS, MEN, and SimLex-999, respectively. In addition, given its upstanding performances in step 1, we add the similarity computed using WUP as an additional criterion for a total of four embedding models selected.

Given that with HSS and WUP we can compute the similarity for each pair of terms in the taxonomy, for this evaluation we have $(n \times (n-1))/2$ pairs of terms for each vector model, where n is the number of words present both in the taxonomy and the text corpus. In our case, the number of common words is 53,451, for a total of 1,428,477,975 possible word pairs for each model, which would make the computation intractable. To reduce the number of samples, we start from 0 pairs and, following [248], we increase the sample size until the Pearson Correlation stabilises. The process of choosing the number of pairs is detailed in Section 6.2.1 section.

Evaluation of the Best Embedding

To evaluate the selected embeddings, we rely on the hypothesis that the embedding that better preserves the semantic knowledge encoded in WordNet will improve the performance of several downstream NLP tasks [46]. Therefore, the performance of the four selected embeddings in the NLP tasks presented in Section 6.1.3 constitutes an extrinsic evaluation of the criteria used to select the vector model, i.e., the similarity measure employed.

Word embeddings can be used as feature vectors of supervised machine learning algorithms used in various NLP tasks. Relying on the hypothesis that unifying lexical and distributional resources in a taxonomy-aware vector model is beneficial for downstream NLP tasks [22, 46, 21], in this section, we examine different queries related to well-known NLP applications.

Categorisation: The categorisation is a natural task to verify to which extent the selected embedding preserves the taxonomic similarity. Since dealing with taxonomic data, this task is especially crucial since having a performant embedding means that the similarity values can reflect the hierarchical relationships among words and preserve the hypernym-hyponym

links. Following the method described in [22], we cluster the vectors from each selected embedding applying a clustering algorithm, and we measure the purity of each cluster. In Section 6.2.3 we describe the process and the metric used to perform this task.

Sentiment classification: Given the strong bond between lexicons and sentiment analysis, for instance, in word polarity disambiguation [305] and in predicting sentiment intensity using stacked ensemble [5], we carry out a sentiment classification task on three customer review datasets, with two binary classifications and one multi-class sentiment classification. Section 6.2.3 describes in detail the datasets used in this task and their characteristics, together with comments on the achieved results.

Hypernym detection: In this task, we examine to what extent the chosen embeddings (see Section 6.2.3) can be utilised to identify the hyponym/hypernym relation between two words in a given pair. In Section 6.2.3 we describe the process for generating the training and test data while commenting on the results of this task on each dataset.

Synonym detection: This task aims to evaluate the impact of the input word embeddings on the performance of a classifier that is trained to detect the synonym pair of words. Section 6.2.3 provides details about the way positive and negative pairs were generated and used to train the classifier.

6.2 Experimental Results

Our experiments rely on a lexical taxonomy and a corpus:

- **Taxonomy:** WordNet [203] provides a structured hierarchy of meanings (senses) and synsets (a collection of words belonging to a specific context). We used the implementation of WordNet inside the NLTK library [27] while calculating the required information both using NLTK's native functions (e.g., calculating the lowest common ancestor) and custom functions (e.g., calculation of cardinality).
- **Corpus:** English Wikipedia dump of the year 2008. The main reason for not choosing a more recent dump is that the last release of WordNet 3.0 (the version we used for our experiments) belongs to 2006, making the use of a newer version of Wikipedia dump

unnecessary. The used dump already includes the pre-processed version of data that we used as our data without performing further cleaning.

6.2.1 Step 1: Evaluation of the HSS

In this section we compare the HSS with the other measures of semantic similarity presented in previous sections through two tasks: *semantic similarity* and *word clustering* (also called *concept categorisation*) introduced in Section6.1.1.

Semantic Similarity

For this task, we consider the Pearson and Spearman's Correlation between the pairwise similarity assessed by the similarity metrics and by humans. The six well-known resources considered in this chapter are MEN, MC30, WSS (the similarity portion of WordSim-353), SimLex-999, MT287, and MT771. Among them, MEN is proposed as a *Semantic Relatedness* dataset, even though it does not distinguish between similarity and relatedness, conflating the two [46]; for this reason, and given its relevance in the literature, we decided to use it for the evaluation together with semantic similarity datasets.

A brief description of the employed datasets is provided in the Table 6.1.

Dataset	Source	Description
MEN	Bruni et al. [38]	3000 random pairs that appeared at least 700 times in ukWaC and Wackypedia corpora with human-assigned similarity judgements, obtained by crowd-sourcing
M30	Miller and Charles [203]	30 pairs selected from the RG65 dataset [244] with the similarity scores obtained from 38 participants
WSS	Agirre et al. [4]	A part of the WordSim353 dataset provided by [85] that addresses only the similarity, unlike the original dataset that has both relatedness and similarity
SimLex-999	Hill et al. [119]	Containing of 111 adjective pairs, 666 nouns pairs and 222 verb pairs
MT287	Radinsky et al. [231]	287 pairs evaluated using human annotators from Mechanical Turk workers
MT771	Halawi et al. [109]	771 word pairs evaluated using human annotators from Mechanical Turk workers, with an average of 20 worker ratings for each word pair

Table 6.1 Descrption of the datasets used in the study

Choosing the number of pairs To define the minimum number of pairs required for our experiments, after randomly generating 100k pairs presented both in WordNet and our corpus vocabulary, we have recursively generated samples of pairs while increasing the number of the samples by 100 in each step. Figure 6.2 shows the Pearson correlation of HSS score and cosine similarity of each paired sample while the orange line indicates the rolling average of 100. To define the Point of Stability (POS) (i.e., a point from which the correlation remains within the POS), we considered a Corridor of Stability (COS) of ± 0.01. Although rules of thumb proposed by [61] and the work of [248] suggest a bigger COS (between 0.05 and 0.1) due to difficulties to achieve a tighter COS mainly caused by the cost of the additional samples. Since such a cost would not be relevant in our case, we decided to consider COS as ± 0.01, which resulted in a POS of 35,000.

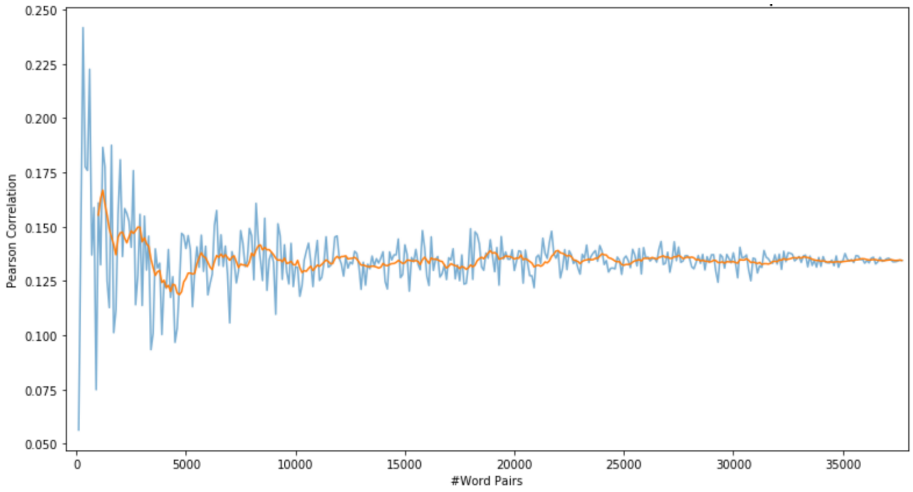


Figure 6.2 Variation of Pearson Correlation Vs. Number of word pairs. Orange line: The rolling average of 100

Comparative results Table 6.2 shows the results of calculating Pearson and Spearman correlation coefficients among six datasets annotated by humans and eight similarity scores. HSS outperforms the other measures (except for the SimLex-999 dataset), both in terms of Pearson and Spearman correlations with the human-annotated datasets. The closest performance to HSS is achieved by the WUP metric. These results confirm the performance superiority of HSS to the SOTA measures (both path-based and information-content-based measures). To conduct this experiment, we used the implementation of WUP, LC, and shortest path in the NLTK library while in the case of Resnik, Jiang-Conrath, Lin, and Seco (Resnik

using Seco Information-content) we implemented them in Python. Table 6.2 also reports the average time for calculating the similarity between two words (in seconds), calculated as the average time in seconds for 1000 random pairs. Our model has the best performance in terms of computational time since it is twice faster as the second-best performer that is the Resnik measure, which uses Information-content calculated by Seco et al. [251].

Word Clustering

To measure the similarity between words belonging to the same concept, we compute the silhouette coefficient of the cluster of words belonging to the same category in human-annotated datasets. The three datasets used are:

- **ESSLLI** (European Summer School in Logic, Language and Information): 45 words divided into 9 categories [23].
- **AP** (Almuhareb and Poesio): 402 words divided into 21 semantic classes [9].
- **BM** (Battig and Montague) 5321 concepts belonging to 56 categories [24].

The silhouette coefficient is a cluster validity measure that compares how similar an object is to its cluster with how similar it is to other clusters. The silhouette is computed as:

$$Silhouette(i) = \frac{b(i) - a(i)}{\max(b(i), a(i))} \tag{6.6}$$

where $a(i)$ is the mean distance between i and all other data points in the same cluster, and $b(i)$ is the smallest mean distance of i to all points in any other cluster. The silhouette ranges from -1 to $+1$: a value near to $+1$ indicates that the object is well matched to its own cluster and poorly matched to neighbouring clusters, and the other way around for a value near to -1 [17].

Comparative results Table 6.3 shows the word clustering task among the three datasets described above and eight similarity scores. `HSS` outperforms all the other measures only LC reaches an equivalent performance in terms of the silhouette on the BM dataset.

These results confirm the performance superiority of the `HSS` with respect to the SOTA measures.

6.2.2 Step 2: A Tool for Computing Semantic Similarity

The python library *TaxoSS* that we created allows the user to easily compute semantic similarity between concepts using eight different measures: HSS, WUP, LC, Shortest Path, Resnik, Jiang-Conrath, Lin, and Seco.

In Figure 6.3 and 6.4 is shown the use of the package for the computation of the semantic similarity between two words using different metrics. In Figure 6.4 is also shown how the user can use her/his corpus in order to compute the similarity through corpus-based similarity measures.

```
from TaxoSS.functions import semantic_similarity
semantic_similarity('brother', 'sister', 'hss')

3.353513521371089
```

Figure 6.3 An example of the use of the semantic similarity *function with the HSS metric.*

```
from TaxoSS.functions import semantic_similarity
semantic_similarity('cat', 'dog', 'resnik')

6.169410755220327

from TaxoSS.calculate_IC import calculate_IC
calculate_IC('data/corpus_test.csv', 'venv/lib/python3.6/site-packages/TaxoSS/data/test_IC.csv')
semantic_similarity('cat', 'dog', 'resnik', 'venv/lib/python3.6/site-packages/TaxoSS/data/test_IC.csv')

3.5077209766856137
```

Figure 6.4 An example of the use of the semantic similarity *function with Resnik metric and the use of an ad hoc Information Content file created through a corpus of choice.*

In Table 6.4 are reported the average times requested for each similarity measure to compute 100 similarities between 100 pairs of words.

6.2.3 Step 3: Embeddings Selection and Evaluation of the Best Embedding

In this section, we generate 80 different embedding models with fastText. Among them, we select the four that better correlates their cosine similarity with the semantic similarity,

measured respectively with the HSS, MEN, SimLex-999, and WUP. To evaluate these four models and compare the semantic similarity measures used to select them, we use their vectors as input features in four downstream NLP tasks. The four NLP tasks are, in order of presentation: Categorisation, Sentiment Classification, Hypernym Detection, and Synonym Detection.

Generation of embeddings We trained our vector model with the fastText library using both *skipgram* and *CBOW*. We tested:

- Five values of the size of the embeddings: 25, 50, 100, 250, and 500;
- Four for the number of epochs: 5, 10, 20, and 50;
- Two for the learning rate: 0.05 and 0.1.

We considered subwords with 5 to 6 letters while setting the minimum word count as 100, running on an Intel Core i7 CPU equipped with 32GB RAM. The best embeddings chosen by each measure for each dataset are reported in the Table 6.5.

Comments on the best embedding To better clarify the matter, using ap [9] dataset, in Figure6.5 we provide a scatter plot produced over the best embedding model - as emerges from Table 6.5 - generated with UMAP[4]. We chose twenty random records for the first ten categories. Each icon and colour is assigned to one category, demonstrating how well the clusters are separated from each other. Analysing the scatter plot one might observe that:

- Categories that are conceptually more distant respect to the other categories show a more clear separation, for instance ***chemical element*** and × ***monetary unit***, while categories semantically close together have a less clear boundaries like ***legal documents*** and ★ ***assets***;
- There are cases that are on the borderline of two or more classes. While such cases may seem the sign of the model weakness, in fact such observations are totally explainable based on the nature of the non-contextual embeddings, since by definition they cannot represent words with multiple meanings. For example, in the scatter plot mentioned above, the word ***capital***, which belongs to ★ ***assets*** in WordNet, is on the border of

[4]Uniform Manifold Approximation and Projection (UMAP) is a dimension reduction technique that can be used for visualisation similarly to t-SNE, but also for general non-linear dimension reduction

♦ ***social unit*** and ***district***. This is could be explained by the fact that we used the Wikipedia corpus (i.e. a generic corpus) for generating the embeddings.

	HSS (ours)		**WUP [304]**		**LC [160]**		**Shortest Path**	
	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman
MEN [38]	**0.41**	**0.33**	0.36	**0.33**	0.14	0.05	0.07	0.03
MC30 [244]	**0.74**	0.69	**0.74**	**0.73**	0.33	0.21	0.22	0.3
WSS [4]	**0.68**	**0.65**	0.58	0.59	0.36	0.23	0.16	0.1
SimLex-999 [119]	0.4	0.38	**0.45**	**0.43**	0.26	0.15	0.2	0.16
MT287 [231]	**0.46**	**0.31**	0.4	0.28	0.26	0.12	0.11	0.11
MT771 [109]	**0.44**	0.4	0.43	**0.49**	0.06	0.02	0.1	0.13
Time per pair (s)	**0.0007**		0.008		0.0055		0.0064	
	Resnik [237]		**Jiang-Conrath [134]**		**Lin [170]**		**Seco [251]**	
	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman	Pearson	Spearman
MEN [38]	0.05	0.03	-0.05	-0.04	0.05	0.04	-0.01	0.03
MC30 [244]	0.13	0.03	-0.06	-0.01	0.05	0.01	0.13	-0.09
WSS [4]	0.02	-0.03	0.04	0.06	0.03	0.06	-0.01	-0.04
SimLex-999 [119]	-0.04	-0.04	0.12	0.14	0.12	0.14	-0.02	-0.08
MT287 [231]	0.03	0.04	0.18	0.16	0.22	0.17	0	-0.06
MT771 [109]	0	-0.01	0	0	0	0	-0.05	-0.03
Time per pair (s)	0.5586		0.551		0.5866		0.0013	

Table 6.2 Semantic similarity

	HSS (ours)	WUP [304]	LC [160]	Shortest Path
ESSLLI [23]	**0.18**	0.11	-0.01	-0.01
AP [9]	**0.33**	0.09	0.04	0.01
BM [24]	**0.09**	0.02	**0.09**	-0.03
	Resnik [237]	Jiang-Conrath [134]	Lin [170]	Seco [251]
ESSLLI [23]	-0.05	-0.37	-0.06	-0.29
AP [9]	-0.09	-0.13	-0.15	-0.32
BM [24]	-0.11	-0.39	-0.19	-0.42

Table 6.3 Cluster purity

	HSS	WUP [304]	LC [160]	Shortest Path
100 pairs	5.77	5.29	5.38	5.29
	Resnik [237]	Jiang-Conrath [134]	Lin [170]	Seco [251]
100 pairs	5.80	5.88	5.98	5.55

Table 6.4 Mean time (seconds) requested for computing the semantic similarity of 100 pairs with `TaxoSS`*.*

	type	**size**	**epoch**	**learning rate**
HSS (ours)	skipgram	500	5	0.05
MEN [38]	skipgram	250	10	0.05
SimLex-999 [119]	CBOW	500	50	0.01
WUP [304]	CBOW	50	10	0.01

Table 6.5 Best embeddings for each measure/dataset

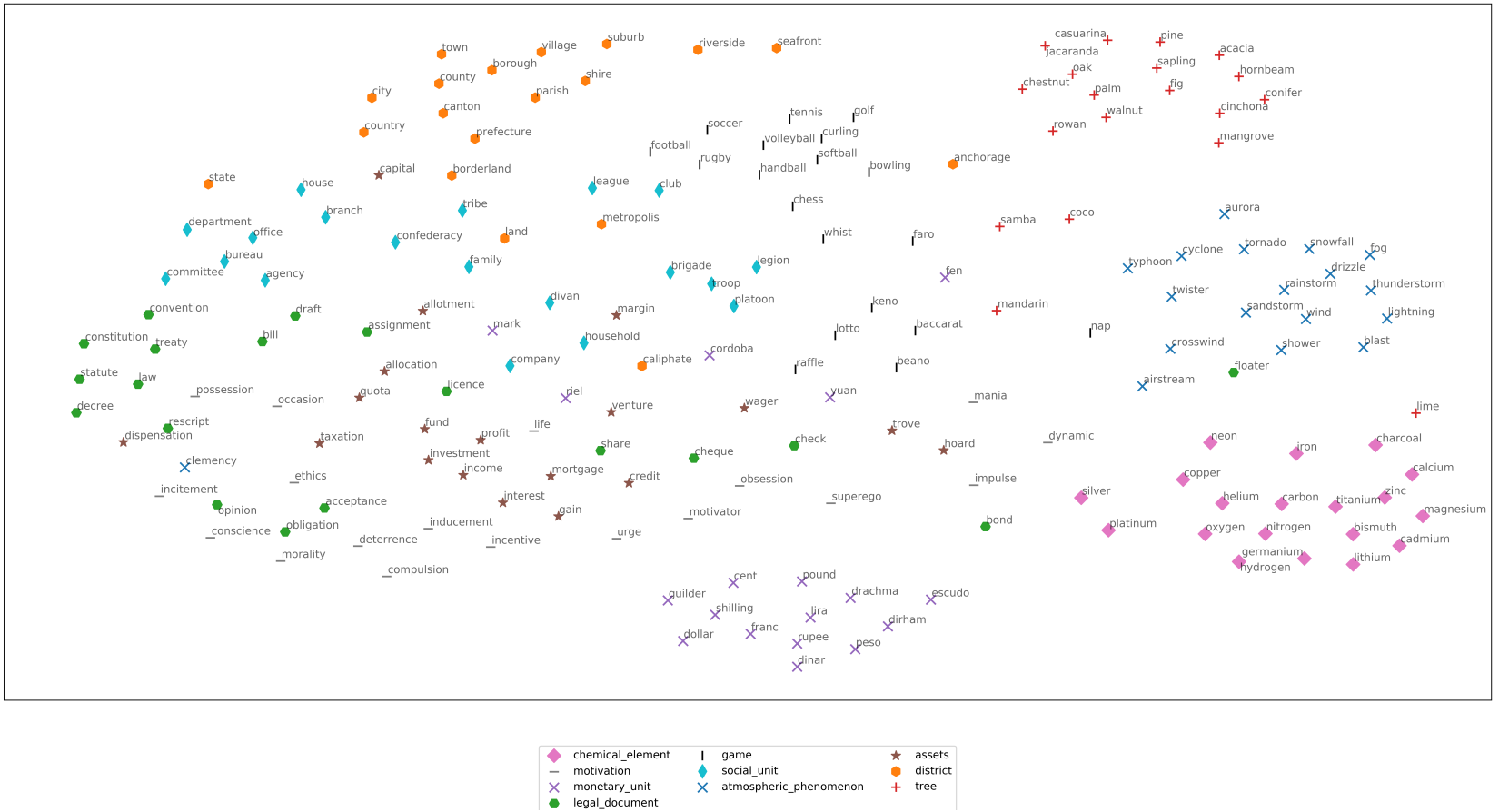


*Figure 6.5 UMAP plot of the **best** word-embedding model resulting from Table 6.5, that is skipgram, dim=500, epochs=5 and learning rate = 0.05. Each icon is assigned to one category in ap [9].*

Below, we describe the four downstream NLP tasks performed to evaluate extrinsically the embedding selection procedure. For each task, we comment on the results.

Categorisation

Following [22] we cluster the vectors from each selected embedding, applying the k-means algorithm from the scikit-learn library [221], with the default parameters. In the next step, the purity of each cluster is calculated. The number of clusters is equal to the number of classes in each dataset. To account for the variation of the results we compute the average value of 50 iterations for each pair.

The datasets used in this experiments are **ap** [9], containing 403 concepts organised into 21 categories, **battig** [24], including 4668 concepts belonging to 56 categories, and **esslli** [23], from the ESSLLI 2008 Distributional Semantic Workshop shared-task set, containing 45 concepts divided into 9 categories.

The purity values of clusters are reported in Table 6.6, where a purity close to 1 shows that the cluster is well reproduced, while a purity close to 0 indicates poor cluster quality. These results show that the embedding chosen by our similarity measure outperforms the other three embeddings, chosen by MEN, SimLex-999, and WUP measure, when applied on three well know categorisation datasets used by [22].

	Categorisation		
	ap [9]	battig [24]	esslli [23]
HSS (ours)	**0.75**	**0.62**	**0.81**
MEN [38]	0.71	0.58	0.77
SimLex-999 [119]	0.71	0.49	0.78
WUP [304]	0.68	0.43	0.73

Table 6.6 Categorisation: Cluster purity obtained from each embedding/dataset

Sentiment classification

We carry out the sentiment classification task on three user review datasets: Binary Movie Review, Binary and Multi-class Amazon Review.

Replicating the experiment done in [247], we perform a binary sentiment classification on the dataset from [181]. This dataset contains 50K movie reviews, divided equally between

binary labels. Utilising the embeddings as the features of a LIBLINEAR logistic regression [81], we compute a linear combination of embeddings, weighted by the word count in each review. We use the scikit-learn library [221] to apply the mentioned regression and calculate the accuracy values for each selected embedding by doing 10-fold cross-validation. The results of this task (classification accuracy) can be seen in Table6.7.

Using the binary and the multi-class Amazon cellphone review datasets[5] we perform both binary and multi-class sentiment classification. These datasets contain $26,845$ and $29,988$ reviews labelled 0 or 1 for the binary dataset and from 1 (absolutely negative) to 5 (absolutely positive) for the multi-class dataset. In both cases, we down-sample the dataset based on the minority label. Table 6.7 shows the mean and the standard deviation of the performance metrics that result from 10-fold cross-validation. Our measure can outperform MEN, SimLex-999, and WUP benchmarks. The exception is the multi-class dataset, in which `HSS` produces results similar to those of SimLex-999.

	Accuracy	**Precision**	**Recall**	**F1**
	Multi-class Amazon Review			
HSS (ours)	**0.4** $\pm$ 0.04	**0.4** $\pm$ 0.04	**0.41** $\pm$ 0.04	**0.4** $\pm$ 0.04
MEN [38]	0.4 $\pm$ 0.04	0.39 $\pm$ 0.04	0.4 $\pm$ 0.04	0.39 $\pm$ 0.04
SimLex-999 [119]	**0.41** $\pm$ 0.02	**0.4** $\pm$ 0.02	**0.41** $\pm$ 0.02	**0.4** $\pm$ 0.02
WUP [304]	0.34 $\pm$ 0.06	0.33 $\pm$ 0.04	0.34 $\pm$ 0.06	0.31 $\pm$ 0.05
	Binary Amazon Review			
HSS (ours)	**0.82** $\pm$ 0.02	**0.81** $\pm$ 0.03	**0.84** $\pm$ 0.03	**0.82** $\pm$ 0.02
MEN [38]	0.81 $\pm$ 0.02	0.79 $\pm$ 0.03	0.83 $\pm$ 0.02	0.81 $\pm$ 0.02
SimLex-999 [119]	0.8 $\pm$ 0.04	0.79 $\pm$ 0.04	0.82 $\pm$ 0.05	0.8 $\pm$ 0.04
WUP [304]	0.72 $\pm$ 0.03	0.71 $\pm$ 0.04	0.74 $\pm$ 0.04	0.72 $\pm$ 0.03
	Binary Movie Review			
HSS (ours)	**0.84** $\pm$ 0.01	**0.84** $\pm$ 0.01	**0.84** $\pm$ 0.02	**0.84** $\pm$ 0.01
MEN [38]	0.82 $\pm$ 0.01	0.82 $\pm$ 0.01	0.82 $\pm$ 0.02	0.82 $\pm$ 0.01
SimLex-999 [119]	0.83 $\pm$ 0.01	0.83 $\pm$ 0.01	**0.84** $\pm$ 0.02	0.83 $\pm$ 0.01
WUP [304]	0.72 $\pm$ 0.02	0.72 $\pm$ 0.02	0.72 $\pm$ 0.02	0.72 $\pm$ 0.02

Table 6.7 Sentiment Classification

[5] https://jmcauley.ucsd.edu/data/amazon/

Hypernym Detection

For this task, we employ the BATS benchmark dataset [100], which contains 99,200 questions in 40 morphological and semantic categories. In particular, we use three lexicography categories, namely *Hypernyms* (Animals, Miscellaneous) and *Hyponyms*(Miscellaneous).

To generate hypernym-hyponym pairs (corresponding to the positive pairs), we extract all the possible pairs, then we deduplicate them, and we remove all the pairs with at least one word out of the embeddings vocabulary. The process leads to 1129 pairs. To generate the negative pairs, we use the aforementioned categories to randomly select words and form pairs, arriving at 1129 pairs that do not have a hypernym-hyponym relationship. Finally, by subtracting the vectors of pair words, we create a single vector, and we use it as the feature for training the classifier. To compensate for the potential effect of the randomly generated pairs on the results, we repeat the process ten times, each time using a logistic regression model to classify the pairs.

Table 6.8 shows the mean and standard deviation of the outcomes for chosen embeddings. The HSS outperforms the other benchmarks except for the MEN dataset, which achieves the same precision as our method.

	Hyper/Hyponym Classification			
	Accuracy	Precision	Recall	F1
HSS (ours)	**0.72** $\pm$ 0.013	**0.72** $\pm$ 0.012	**0.75** $\pm$ 0.017	**0.73** $\pm$ 0.013
MEN [38]	0.71 $\pm$ 0.014	**0.72** $\pm$ 0.017	0.72 $\pm$ 0.015	0.72 $\pm$ 0.012
SimLex-999 [119]	0.69 $\pm$ 0.023	0.68 $\pm$ 0.251	0.73 $\pm$ 0.017	0.70 $\pm$ 0.019
WUP [304]	0.68 $\pm$ 0.015	0.71 $\pm$ 0.02	0.65 $\pm$ 0.014	0.68 $\pm$ 0.013

Table 6.8 Hypernym detection

Synonym Detection

As for the previous task, we use the BATS benchmark to generate the training data for the logistic regression classifier. To create positive pairs (i.e. pairs with synonym relationship), first, we generate all the possible combinations of 2 for each entry in *Synonym-exact* and *Synonym-intensity* files. Then, we combine them with the pair made from the synonym words in the *Antonym-gradable* file, which results in 4,663 pairs that we reduced to 4,011 pairs after

deduplication. Similar to the method described in the previous task, we generate negative pairs by iterating through the vocabulary ten times.

As reported in Table 6.9, despite close results to the SimLex-999 dataset, the embedding chosen by the HSS carries out a better synonym classification than the other methods.

	Synonym Classification			
	Accuracy	Precision	Recall	F1
HSS (ours)	**0.616** $\pm$ 0.006	**0.609** $\pm$ 0.007	**0.624** $\pm$ 0.012	**0.617** $\pm$ 0.006
MEN [38]	0.586 $\pm$ 0.006	0.58 $\pm$ 0.007	0.586 $\pm$ 0.008	0.583 $\pm$ 0.006
SimLex-999 [119]	0.612 $\pm$ 0.007	0.605 $\pm$ 0.008	0.618 $\pm$ 0.004	0.611 $\pm$ 0.005
WUP [304]	0.54 $\pm$ 0.007	0.534 $\pm$ 0.007	0.537 $\pm$ 0.014	0.536 $\pm$ 0.007

Table 6.9 Synonym detection

7

Taxonomy Refinement Through Embedding Evaluation

Taxonomies provide a structured representation of semantic relations between lexical terms. In the case of standard official taxonomies, the refinement task consists of maintaining them updated over time while preserving their original structure. To date, most of the approaches for automated taxonomy refinement rely on word vector models. However, none of them considers to what extent those models encode the taxonomic similarity between words. Motivated by this, in this chapter, we propose and implement `TaxoRef`, a methodology that uses the best embedding resulting from `TaxoVec` methodology (See Chapter 6), to provide a list of possible refinements. Finally, this list is reviewed by domain experts to make the final decisions and choose the valid modifications to be applied to the taxonomy. Figure 7.1 depicts the whole process discuss in Chapter 6 and the current chapter that are Embedding evaluation and Taxonomy refinement using HITL paradigm.

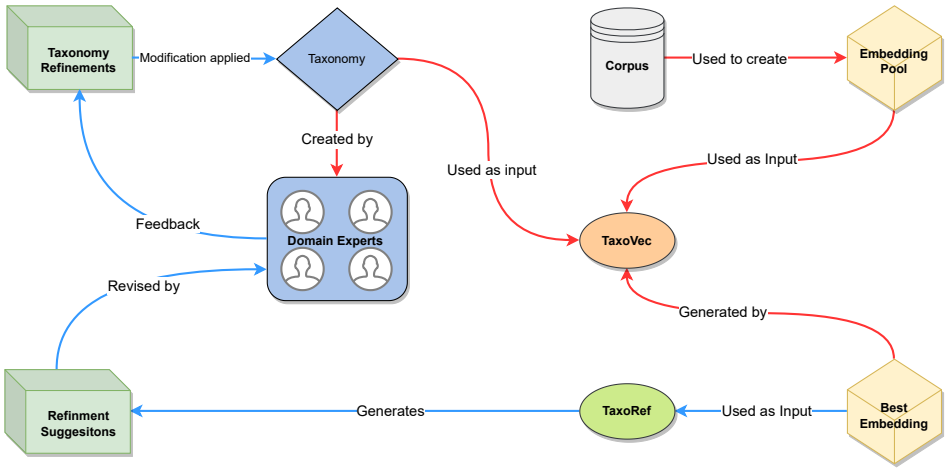


Figure 7.1 A general overview of the integration of human-in-the-loop approach in Taxonomy Refinement task. The red arrows show the embedding evaluation (see 6) while the blue arrows belong to the Taxonomy refinement discussed in this chapter

7.1 The TaxoRef Approach

Below we present TaxoRef, as shown in Figure 7.1. By doing so, we present the criteria for moving a taxonomic entity to a different concept by applying the Bayes theorem. Hence, the probability that the word w, represented in the embedding space by the vector $\mathbf{v}$, belongs to the concept c is given by:

$$p(c \mid \mathbf{v}) = \frac{p(\mathbf{v} \mid c)\, p(c)}{p(\mathbf{v})} \tag{7.1}$$

Thus the word w is assigned to the class c_i iff:

$$p(\mathbf{v} \mid c = c_i)\, p(c = c_i) \geq p\left(\mathbf{v} \mid c = c_j\right) p\left(c = c_j\right), c_j \in C \setminus c_i \tag{7.2}$$

Where the prior probability $p(c)$ is estimated through class frequency and the likelihood $p(\mathbf{v} \mid c)$, is:

$$p(\mathbf{v} \mid c) = p(v_1 \mid c) \times p(v_2 \mid c) \times \ldots \times p(v_D \mid c)$$

where we assume conditional independence between the elements $v_1, v_2, \ldots, v_D$ of the vector $\mathbf{v}$, analogously to the Naive Bayes classifier. The probability $p(v_i \mid c)$ is estimated by a Gaussian density function for $\forall i \in 1, 2, \ldots, D$.

Experimental Settings for Reproducibility. All the experiments have been performed over an Intel i7 processor with Ubuntu-20, equipped with 32GB RAM. `TaxoRef` is implemented with Python 3.7. Tuning parameters are reported for each experiment while the source code of `TaxoRef` is provided on Github[1].

7.2 Experimental Results on 2M+ UK Online Job Ads

While on the one side, ESCO allows comparing different labour markets, on the other side, it does not encode the characteristics and peculiarities of such labour markets in terms of skills advertised - which vary country by country - and the meaning of terms that are used differently based on the level of maturity of local labour markets. `TaxoRef` would allow encoding semantic similarities as they emerge from OJVs (i.e., the labour market demand) within ESCO, identifying relationships that might refine ESCO to fit the labour market demand better. In [93], we used the approach employed by `TaxoRef` to identify novel occupations to enrich the ESCO taxonomy with new emerging jobs.

Data Overview. Our experiments rely on the use of a large corpus of OJVs collected from online sources within the project [12]. We selected the titles of all 2,119,493 online job vacancies referring to ICT jobs in the United Kingdom during the year 2018. Concepts belonging to the fifth and highest level of ESCO (*c5*) are called *narrower labels*, while all the hyponyms of the same narrow label, called *alternative labels*, are co-hyponyms[3] and are different terms which express the same kind of occupation. As we can see in Figure 7.2, the ISCO classification assigns a code only to the first four levels (*c4*). To evaluate the similarity between narrower and alternative labels, we assigned a new code to each narrower label. For instance, if the concept *2512*, at the fourth level of ESCO has two narrower labels as hyponyms, their codes will be 2512_01, and 2512_02 respectively.

Generation of embeddings. We trained our vector model with the fastText library using both *skipgram* and *cbow*. We tested five values of the size of the embeddings (5, 25, 100, 300, 500), five for the number of epochs (10, 25, 100, 200, 300) and four for the learning

[1] https://github.com/Crisp-Unimib/TaxoRef

[2] Preliminary results at https://tinyurl.com/skillovate

[3] co-hyponyms refer to hierarchical concepts which share the same hypernym

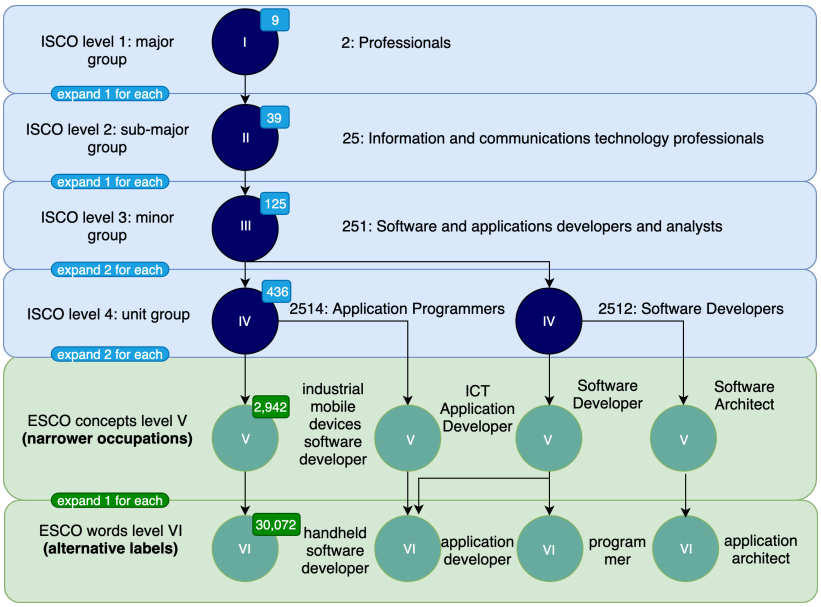


Figure 7.2 The ESCO taxonomy built on top of ISCO

rate (0.05, 0.1, 0.2, 0.5) for a total of 600 embeddings. All the subwords with 3 to 10 letters were considered. Average training times (with std) in seconds were 246 ± 333.

Table 7.1 The fraction of ESCO V and VI level words to be assigned to each ESCO IV level concept according to the criterion in Section 7.1. The rows represent ESCO IV concepts. For each concept (row), the column Accordance reports the fraction of occupations terms which are assigned to the same concept by Eq. 7.2, while the column Refinement shows the fraction of occupation terms assigned to a different ESCO IV level concept, which is specified (note that only the main ones are presented, and rounded to the second decimal place, thus not all the rows sum to 1). Missing concepts do not need a refinement (i.e., Accordance=1)

ESCO Concept	Accordance	Refinement (fraction)				
1330 ICT service managers	1330 (0.5)	2511 (0.18)	2519 (0.12)	2514 (0.06)	2513 (0.06)	3512 (0.06)
2511 Systems analysts	2511 (0.77)	2512 (0.08)	1330 (0.05)	2521 (0.03)	2513 (0.03)	2522 (0.03)
2512 SW developers	2512 (0.82)	2514 (0.11)	2511 (0.05)			
2513 Web and multimedia developers	2513 (0.85)	2512 (0.14)				
2519 SW & Application Developers	2519 (0.9)	2514 (0.1)				
2521 DB designers and admin	2521 (0.66)	2511 (0.33)				
2529 DB and network professionals	2529 (0.69)	2511 (0.15)	2522 (0.08)	2519 (0.08)		

Using the `TaxoVec` method described in Chapter 6 we determine the best embedding made by following hyperparameters: Training mode: CBOW, dimension: 100, epochs: 200 and learning rate: 0.1 (To avoid redundancy, the process is of determining this specific embedding is not described in this section - please see Chapter 6 for details)

7.2.1 Result Comments

Comments on the best embedding.

As discussed previously, the evaluation and selection of the best model are mandatory activities that affect the trustworthiness of all the tasks that use such an embedding as input. To better clarify the matter, in Figure 7.3 we provide a scatter plot produced over the best embedding model generated utilizing UMAP. Each icon is assigned to one ISCO level 4 group, as in Figure 7.2. The ESCO concepts and words belonging to each group are shown, distinguishing between narrower occupations (shallow shape) and alternative labels (filled shape). The embedding shown in Figure 7.3 encodes the occupations as they emerge from the UK labour market (2M+ OJVs in 2018) within the ESCO taxonomy. This is beneficial for labour market specialists as a way to understand and monitor labour market dynamics. Specifically, one might observe that though a *data engineer* and a *data scientist* were designed to be co-hyponyms in ESCO, as they belong both to the *2511: Systems Analysts* ISCO group, their meaning is quite different in the real-labour market, as any computer scientist knows. The former indeed share much more with *2521: Database designers and administrators* rather than its theoretical group. Still along this line, an *IT security technician* seems to be also co-hyponym with occupations in *2529: Database and network professionals*, much more than with terms in its class as specified within ESCO, that is *3512: Information and communications technology user support technicians*. On the other side, one might note that in many cases, the taxonomy perfectly adheres to the real labour market demand for occupations. This is the case of *3521: Broadcasting and audio-visual technicians*, which composes a very tight cluster in the map, showing a perfect match between de-facto and de-jure categories. This also applies to $*$ *3513: Computer network and systems technicians*, even though to a lesser extent. This analysis is useful to labour market specialists and policymakers to identify mismatches in the taxonomies and to provide accurate feedback to improve the taxonomy as well.

At https://tinyurl.com/scatter-umap is available, the UMAP of the best and worst (lower correlation with the `HSS`described in Chapter 6) embeddings. The comparison of the two gives a glance at the benefit deriving from the selection of the best embedding through the `HSS` criterion.

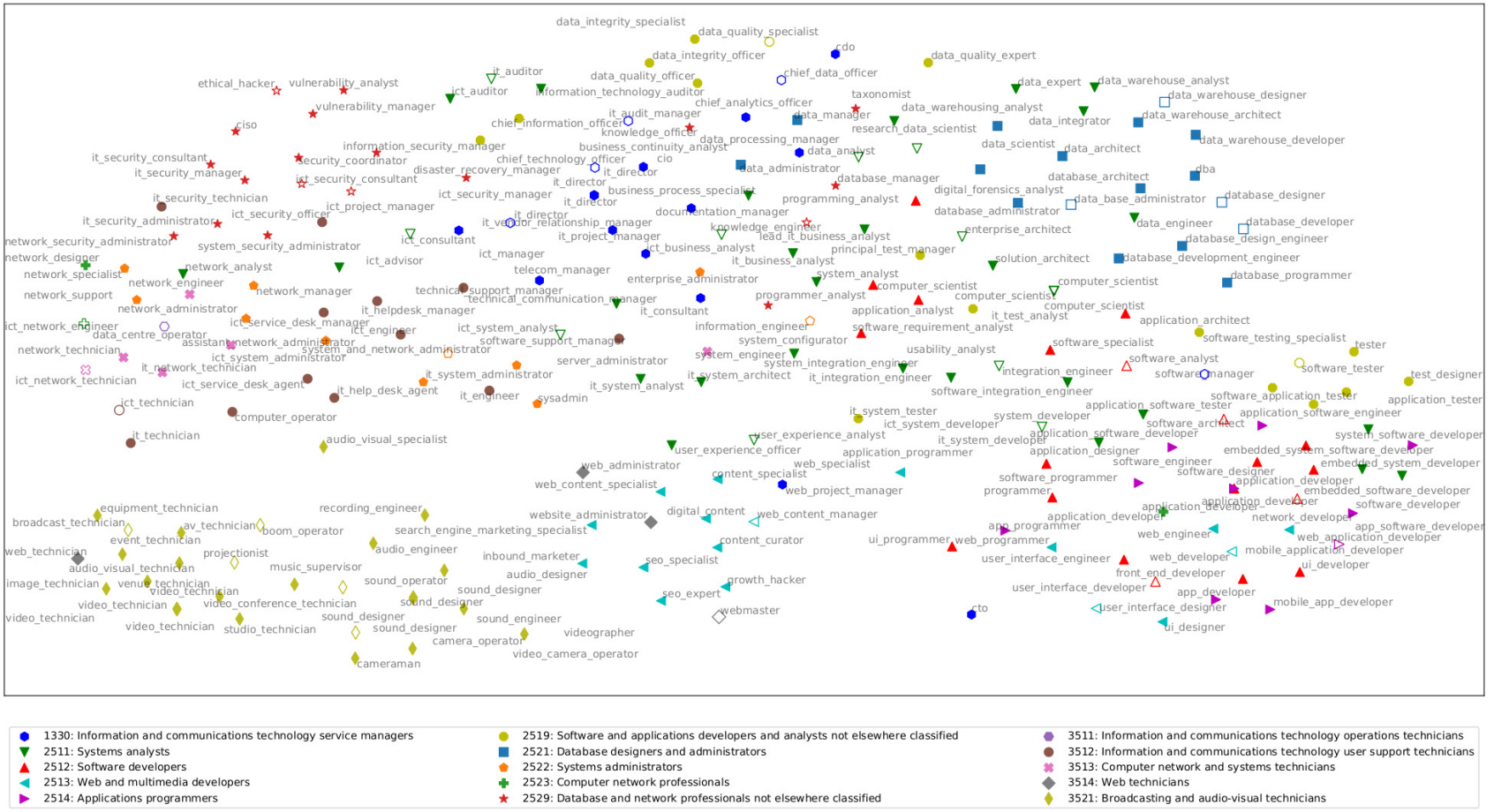


*Figure 7.3 UMAP plot of the **best** word-embedding model resulting from the best model (CBOW dim=100, epochs=200 and learning rate = 0.1). Each icon is assigned to one ISCO level 4 group, as in Figure 7.2. The ESCO concepts and words belonging to each group are shown, distinguishing between narrower occupations (shallow shapes) and alternative labels (filled shapes). Available at higher resolution at https://tinyurl.com/scatter-umap*

Refinement Results on UK-2018.

Standard taxonomies are useful as they represent a lingua franca for knowledge-sharing in many domains. However, as they are built periodically by a panel of experts following a top-down approach, they quickly become obsolete, losing their ability to represent the underlying domain. This is why there is a growing number of attempts to refine taxonomies following a data-driven paradigm. In this section, we employ the methodology described above to find the most suitable concept for each taxonomic entity and suggest these refinements to the LMI experts.

Applying Eq. 7.2, we find that for the 83.4% of the words w analysed, the concept c_i with the highest probability of being its hypernym is its current hypernym in ESCO. In Table 7.1, we present the evaluation for each ICT ESCO concept, excluding those with less than 10 entities. For instance, we can see that all the occupations of the group *3521: Broadcasting and audio-visual technicians* are tightly related among themselves, and none of them is moved to a different group. Conversely, only 50% of the occupations in group *1330: ICT Service Managers* are assigned to the group *1330* itself. For instance, the web project manager is assigned to the class ◀ *2513: Web and multimedia developers*. Note that these results depend on the corpus chosen for the embeddings generation. In this case, we choose ICT-related OJVs posted in 2018 in the UK.

Figure 7.4 shows the refinement proposed by `TaxoRef` for the occupations belonging to the ESCO concept *2511 - Systems Analysts*. For the 77% (28 over 36) of the occupations, the ESCO taxonomy and `TaxoRef` are in accordance, assigning them to the class *2511 - Systems Analysts*. For the remaining eight occupations, `TaxoRef` suggests a different classification. The occupation *user experience officer*, for instance, is reassigned to the ESCO class *2513 - Web and Multimedia Developers*, the *data Engineer* to the class *2521 - DB Designers and Administrators* and so on. All the suggestions for the refinement are highly plausible and can constitute the basis for a discussion among experts on the accordance between the *de jure* taxonomy ESCO and the *de facto* labour market in a specific context as it emerges from labour market demand (OJVs).

As the last part of the `TaxoRef` method, As can be seen in the Table 7.2, we proposed the refinement suggestions to ten LMI experts with different levels of experience with ESCO

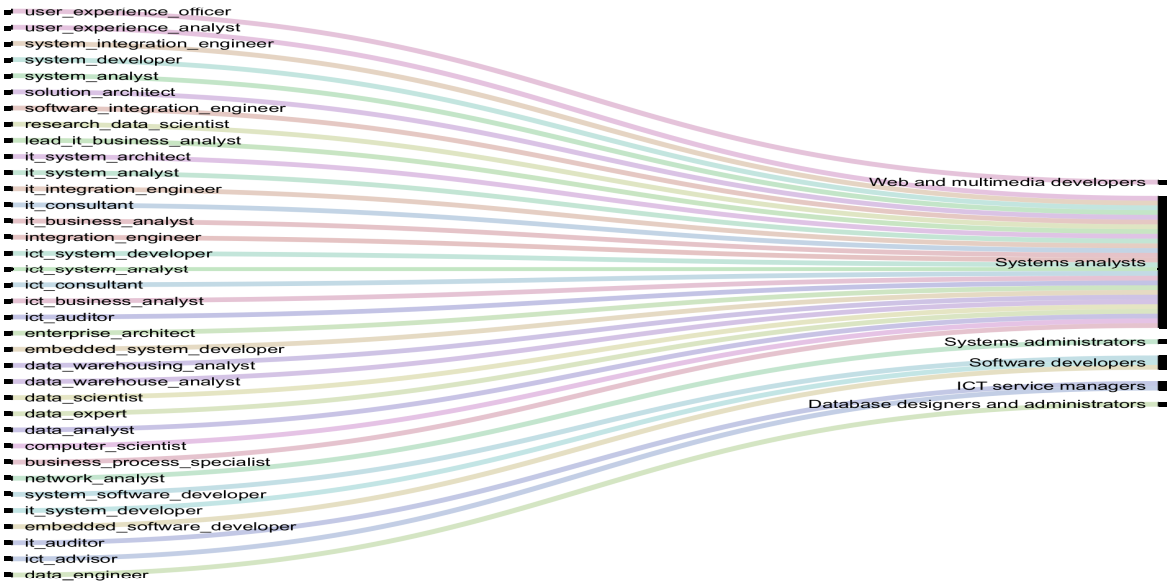


Figure 7.4 Example of refinement for the ESCO concept 2511 Systems Analysts

taxonomy. Each expert, apart from choosing the group they think fits the title better, declared their level of experience regarding ESCO taxonomy [4].

Table 7.2 The titles that were provided to LMI experts int he refinement revision step

Title	**ESCO Classification**	**`TaxoRef` Suggestion**
information engineer	Database and network professionals not elsewhe...	Systems analysts
principal test manager	Software and applications developers and analy...	Systems analysts
application developer	Applications programmers	Software developers
application software developer	Software developers	Applications programmers
embedded software developer	Systems analysts	Applications programmers
system software developer	Systems analysts	Applications programmers
web application developer	Web and multimedia developers	Applications programmers
application software tester	Software and applications developers and analy...	Applications programmers
data engineer	Systems analysts	Database designers and administrators
data warehouse analyst	Systems analysts	Database designers and administrators
user experience officer	Systems analysts	Web and multimedia developers

These experience levels (from extremely poor to extremely high) define the impact weight of each expert's opinion on the final judgement; by *extremely poor* having zero impact. Table 7.3 demonstrates the number of experts for each experience level and their contributions to the final judgements. By summing the multiplication of number of experts and the assigned weights in the Table 7.3 we can calculate the maximum rate each title assignment (i.e. done by ESCO or `TaxoRef`) as 23. Figure 7.5 show the weighted ranks to each title (see Table

[4]The questionary can be find at following link: https://bit.ly/3V7CbHr

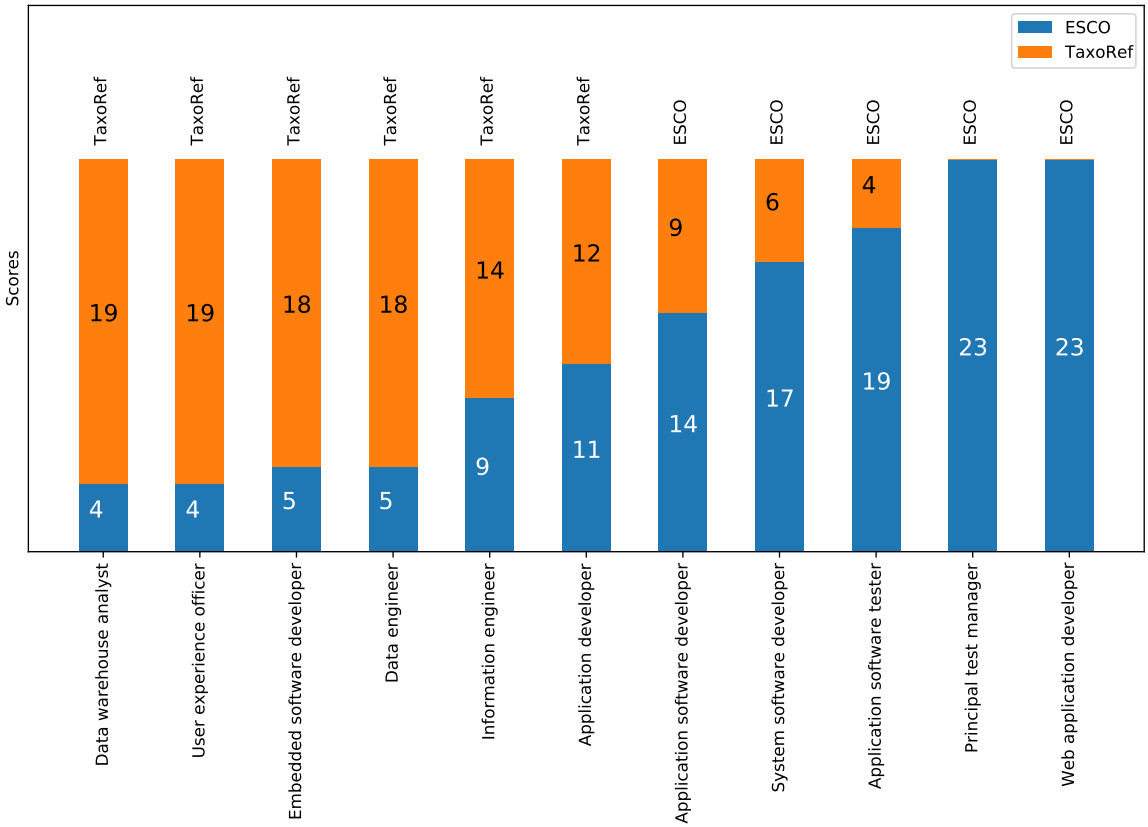


Figure 7.5 The given rank given by experts to suggested refinements. The name written on top of each bar indicates the approval of the refinement (`TaxoRef`) or its rejection (ESCO) by experts

7.2). As the results of this phase, we observe that 6 out of 11 refinemtn suggesions provided by `TaxoRef` are verified by LMI experts who have a high level of experience with ESCO taxonomy. Such results confirm the power of `TaxoRef` to suggest realistic refinements which are aligned with experts' opinions.

Table 7.3 Number of experts for each experience level and their contributions to the final judgements

Experience Level	Number of Respondents	Weight
Exteremly Low	1	0
Low	0	1
Average	6	2
High	1	3
Exteremly High	2	4

8

Conclusion

In this thesis, we explored the integration of Human-in-the-Loop methods with XAI and LMI fields in order to change the role of human from a passive part of the system to an active element of the system that directly impacts the output generated by the system. To do so, we propose novel methods which address conversational explanation generation and taxonomy refinement.

In the first part, we demonstrated why natural language explanations are needed within XAI, how such explanations differ from graphical/numerical presentation methods and what benefits they provide. Further, we proposed the `ConvXAI` system, which uses the HITL paradigm to create a conversational explanation system that can provide explanations for any given state-of-the-art explainer by providing a tool to users in order to request their desired explanations, ask for clarifications and modify the ML pipeline if needed. The novelty of this system, apart from the generation of model-agnostic and contextual explanations, is that by putting the user in a centric role, it is able to modify the model and its hyperparameters in order to improve the quality of final results. In this way, the final user is not merely the final consumer of the output but can decide the way in which the explanations are generated and presented.

Finally, in part II, we study the integration of HITL in LMI. In the beginning chapters of part II, we introduce `HSS` , a semantic similarity metric and `TaxoVec` , an embedding evaluation method that utilizes this metric. In the last chapter of this part, we introduce `TaxoRef` , which addresses one of the most crucial tasks in LMI, i.e. taxonomy refinement. To achieve this goal, `TaxoRef` uses the output of `TaxoVec` as the set of word vectors that best represent a taxonomy-bonded document corpus. By doing so, `TaxoRef` generates a list of modifications, which is reviewed by a group of domain experts to choose the possible changes and refine the taxonomy accordingly. Such an approach drastically improves the quality of the generated taxonomy by incorporating the priceless human knowledge and experience in the taxonomy, which otherwise would be costly, if not impossible, to fully integrate with the generated taxonomy.

8.1 Future Works

Our future research activity is moving on to two contexts.

In the context of conversational explanations, we are working on extending the impact of user feedback to data preparation and pre-processing phases. In this way, the user will be able not only to ask questions about the outputs and explanations but also can investigate the possible biases and data preparation and sampling methods used before the application of the black box model.

We should also mention that while the directions for future works mentioned above will bring significant improvement to XAI systems, achieving them requires a bold modification of the proposed architecture. A possible way to overcome this issue is to integrate the current intent and entity extraction method with Large Language Models (LLMs) in order to reduce the amount of required training data.

As for the taxonomy refinement and embedding evaluation, we are currently investigating the possibility of integrating of HITL approach also in the embedding evaluation in order to extend the role of human experts from taxonomy refinement to embedding evaluation.

9

Acronyms

ADF Agent Dialogue Framework

AHP Analytic Hierarchy Process

ANN Artificial Neural Networks

ASR Automatic Speech Recognition

BDD Binary Decision Diagram

BERT Bidirectional Encoder Representations from Transformers

bi-GRU Bidirectional Gated Recurrent Unit

BLSTM Bidirectional Long Short-term Memory

CI Conversation Initialiser

CNN Convolutional Neural Networks

COS Corridor of Stability

CRF Conditional Random Fields

DL Deep Learning

DP Dialogue Policy

DSM Distributional Semantic Models

DST Dialogue State Tracking

EDM Explanation Dialogue Model

EG Explanation Generator

FIN Feature Inspection

GDPR General Data Protection Regulation

GRU Gated Recurrent Unit

HITL Human-in-the-Loop

ICE Individual Conditional Expectation

ICT Information and Communication Technology

ILO International Standard Organization

ILP Inductive Logic Programming

JSON JavaScript Object Notation

LFIT Learning from Interpretation Transition

LRP Layer-wise Relevance Propagation

ML Machine Learning

MLP Multilayer Perceptron

NB Naive Bayes

NER Named Entity Recognition

NL Natural Language

NLE Natural Language Explanations

NLG Natural Language Generation

NLP Natural Language Processing

NLU Natural Language Understanding

NN Neural Network

PDP Partial dependency

POS Point of Stability

RNN Recurrent Neural Network

RecNN Recursive Neural Network

SLU Spoken Language Understanding

SVM Support Vector Machines

Seq2Seq Sequence-to-sequence

TriCRF Triangular CRF

UML Unified Modeling Language

XAI Explainable Artificial Intelligence

List of Figures

1.1 The development cycle of model [303] 4
1.2 Human-in-the-Loop and Explainable AI in the Future of Work [272] 5
1.3 Example of explanations provided by SHAP and LIME explainers 9

2.1 A roadmap for selecting XAI systems that make use of natural language explanations 20
2.2 An example of a bar plot from [238] 28
2.3 An example of a line plot from [3] 28
2.4 An example of a tree from [194] 29
2.5 An example of a heatmap from [147] 29
2.6 An example of a histogram from [20] 30
2.7 An example of a histogram from [20] 30
2.8 An example of a bubble plot from [282] 31
2.9 An example of a heatmap from [315] 31
2.10 An example of a saliency masking from [255] 32
2.11 An example of an image manipulation from [282] 32
2.12 An example of a tabular report from [268] 33
2.13 An example of a decision table from [281] 33
2.14 An example of a graphical table report from [146] 34
2.15 An example of a decision table from [163] 34
2.16 The AHP hierarchy built from Tab.2.1 weighted by a user. Any user can contribute weighting the hierarchy at https://tinyurl.com/XAI-NLG-AHP . 40
2.17 The paper ranking based on the hierarchy shown in Figure 2.1 41
2.18 Hierarchy (A) and paper ranking (B) of case 1 42

2.19 Hierarchy (A) and paper ranking (B) of case 2 43
2.20 Hierarchy (A) and paper ranking (B) of case 3 44

3.1 Overview of ContrXT, taken from [188] 47
3.2 Indicators for the changes in classification paths from t_1 to t_2 for each *20newsgroup* class. On the x-axis, we present the classification classes, and on the y-axis the ADD/DEL indicators 50
3.3 NLE for *alt.atheism* using the BERT model of Table 3.1 51

4.1 Above diagram: A classic ML-XAI system; Bottom diagram: ConvXAI, capable of generating conversational explanations in which the user contributes by challenging the model and giving feedback on the results. Such feedback then are used to suggest modifications of the model generation phase 54
4.2 UML state transition diagram of the proposed extended Explanation Dialogue Model that enhances the work of [182] through clarification dialogue type. Locutions starting with Q and E refer to the Questioner and the Explainer, respectively. 59
4.3 Components of ConvXAI - Numbers on arrows show the sequence of actions: 1)user interacts with conversation initialiser 2)conversation initialiser communicates with DST 3)DST uses NLU 4)NLU passes its output to DST 5)DST passes the sate to DP 6) DP either directly answers the user or communicate with explanation generator 7)explanation generator interacts with the user . 61
4.4 Example of Conversation Initializer 72
4.5 Example of a contrastive explanatin 72
4.6 Example of a global explanation 73
4.7 Example of how ConvXAI handles user's intention change 73
4.8 Example of how ConvXAI handles user's non-logical input 74
4.9 Example of a *why* explanation 74
4.10 User Study: SHAP - Plot 4 77
4.11 User Study: SHAP - Plot 3 78
4.12 User Study: LIME - Plot 2 78
4.13 User Study: SHAP - Plot 1 79

4.14 User Study: SHAP - Plot 2 79
4.15 User Study: LIME - Plot 1 79
4.16 How groups evaluated Adequacy of explanations - The red X shows the mode of values 81
4.17 How groups evaluated the usefulness of explanations for discovering the causality of explanations - The red X shows the mode of values 82
4.18 How groups evaluated the independence level in comprehending explanations - The red X shows the mode of values 82
4.19 How groups evaluated the independence level in comprehending explanations - The red X shows the mode of values 83
4.20 How groups evaluated how comprehensive are the explanations - The red X shows the mode of values 83
4.21 How technical group evaluated Adequacy of explanations - The red X shows the mode of values 84
4.22 How technical group evaluated the usefulness of explanations for discovering the causality of explanations - The red X shows the mode of values 84
4.23 How technical group evaluated the comprehension ease of explanations - The red X shows the mode of values 85
4.24 How technical group evaluated the independence level in comprehending explanations - The red X shows the mode of values 85
4.25 How technical group evaluated how comprehensive are the explanations - The red X shows the mode of values 86
4.26 Clarifications scores given to Plot 2 generated by LIME explainer - The red X shows the mode of values 87
4.27 Clarifications scores given to Plot 1 generated by SHAP explainer - The red X shows the mode of values 88
4.28 Clarifications scores given to Plot 4 generated by SHAP explainer - The red X shows the mode of values 88
4.29 Clarifications scores given to Plot 4 generated by SHAP explainer - The red X shows the mode of values 89

4.30 The overall scores given by groups to dialogues - The red X shows the mode of values 89

6.1 A general overview of the integration of human-in-the-loop approach in Embedding evaluation task 106
6.2 Variation of Pearson Correlation Vs. Number of word pairs. Orange line: The rolling average of 100 113
6.3 An example of the use of the *semantic similarity* function with the `HSS` metric. 115
6.4 An example of the use of the *semantic similarity* function with Resnik metric and the use of an ad hoc Information Content file created through a corpus of choice. 115
6.5 UMAP plot of the **best** word-embedding model resulting from Table 6.5, that is *skipgram, dim=500, epochs=5 and learning rate = 0.05*. Each icon is assigned to one category in ap [9]. 119

7.1 A general overview of the integration of human-in-the-loop approach in Taxonomy Refinement task. The red arrows show the embedding evaluation (see 6) while the blue arrows belong to the Taxonomy refinement discussed in this chapter 126
7.2 The ESCO taxonomy built on top of ISCO 128
7.3 UMAP plot of the **best** word-embedding model resulting from the best model (*CBOW dim=100, epochs=200 and learning rate = 0.1*). Each icon is assigned to one ISCO level 4 group, as in Figure 7.2. The ESCO concepts and words belonging to each group are shown, distinguishing between narrower occupations (shallow shapes) and alternative labels (filled shapes). Available at higher resolution at https://tinyurl.com/scatter-umap 130
7.4 Example of refinement for the ESCO concept *2511 Systems Analysts* 132
7.5 The given rank given by experts to suggested refinements. The name written on top of each bar indicates the approval of the refinement (`TaxoRef`) or its rejection (ESCO) by experts 133

List of Tables

2.1 Mapping selected papers to our roadmap. *(Example and Benchmark)* → Not provided/used: □, Provided/used once: ⧠, Provided/used multiple times: ■ ; *(Dataset)* → Not mentioned: 👁‍🗨, Private dataset: 🔒, Public dataset: 🔓; *(Code)* → Not provided: 🔑, Provided no documentation: git, Provided with documentation: git; *(Rest of features)* → Not mentioned: ○, Mentioned but not applied: ◉, Applied: ● 37

3.1 ContrXT on 20newgroups (D_{t_1}, D_{t_2} from [135]) varying the ML algorithm. • indicates the best surrogate. 50

4.1 Final list of explanation-related intents 66
4.2 Macro-structure of dialogues used in the user study 77
4.3 Groups' pair-wise Krippendorff's alpha for each question 81

6.1 Descrption of the datasets used in the study 112
6.2 Semantic similarity 118
6.3 Cluster purity 118
6.4 Mean time (seconds) requested for computing the semantic similarity of 100 pairs with TaxoSS. 118
6.5 Best embeddings for each measure/dataset 118
6.6 Categorisation: Cluster purity obtained from each embedding/dataset 120
6.7 Sentiment Classification 121
6.8 Hypernym detection 122
6.9 Synonym detection 123

7.1 The fraction of ESCO V and VI level words to be assigned to each ESCO IV level concept according to the criterion in Section 7.1. The rows represent ESCO IV concepts. For each concept (row), the column *Accordance* reports the fraction of occupations terms which are assigned to the same concept by Eq. 7.2, while the column *Refinement* shows the fraction of occupation terms assigned to a different ESCO IV level concept, which is specified (note that only the main ones are presented, and rounded to the second decimal place, thus not all the rows sum to 1). Missing concepts do not need a refinement (i.e., Accordance=1) 128

7.2 The titles that were provided to LMI experts int he refinement revision step 132

7.3 Number of experts for each experience level and their contributions to the final judgements 133

Bibliography

[1] Abdul, A., Vermeulen, J., Wang, D., Lim, B. Y., and Kankanhalli, M. (2018). Trends and trajectories for explainable, accountable and intelligible systems: An hci research agenda. In *CHI*.

[2] Adams, D. (1984). *Life, the Universe and Everything: Hitchhiker's Guide to the Galaxy Book 3*, volume 3. Tor UK.

[3] Adler, P., Falk, C., Friedler, S. A., Nix, T., Rybeck, G., Scheidegger, C., Smith, B., and Venkatasubramanian, S. (2018). Auditing black-box models for indirect influence. *Knowledge and Information Systems*, 54(1):95–122.

[4] Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., and Soroa, A. (2009). A study on similarity and relatedness using distributional and wordnet-based approaches.

[5] Akhtar, M. S., Ekbal, A., and Cambria, E. (2020). How intense are you? predicting intensities of emotions and sentiments using stacked ensemble. *IEEE CIM*, 15(1).

[6] Akula, A. R., Todorovic, S., Chai, J. Y., and Zhu, S.-C. (2019). Natural language interaction with explainable ai models. In *CVPR*.

[7] Alabdulkareem, A., Frank, M. R., Sun, L., AlShebli, B., Hidalgo, C., and Rahwan, I. (2018). Unpacking the polarization of workplace skills. *Science advances*, 4(7).

[8] AlMousa, M., Benlamri, R., and Khoury, R. (2021). Exploiting non-taxonomic relations for measuring semantic similarity and relatedness in wordnet. *Knowledge-Based Systems*, 212:106565.

[9] Almuhareb, A. (2006). *Attributes in lexical acquisition*. PhD thesis, University of Essex.

[10] Alonso, J. M. and Bugarín, A. (2019). Expliclas: automatic generation of explanations in natural language for weka classifiers. In *FUZZ-IEEE*. IEEE.

[11] Alonso, J. M., Ramos-Soto, A., Reiter, E., and van Deemter, K. (2017). An exploratory study on the benefits of using natural language for explaining fuzzy rule-based systems. In *FUZZ-IEEE*.

[12] Aly, R., Acharya, S., Ossa, A., Köhn, A., Biemann, C., and Panchenko, A. (2019). Every child should have parents: a taxonomy refinement algorithm based on hyperbolic term embeddings. *arXiv preprint arXiv:1906.02002*.

[13] Amarasinghe, K. and Manic, M. (2019). Explaining what a neural network has learned: Toward transparent classification. In *FUZZ-IEEE*. IEEE.

[14] Androutsopoulos, I. and Malakasiotis, P. (2010). A survey of paraphrasing and textual entailment methods. *JAIR*, 38.

[15] Aouicha, M. B., Taieb, M. A. H., and Hamadou, A. B. (2018). Sisr: System for integrating semantic relatedness and similarity measures. *Soft Computing*, 22(6).

[16] Apicella, A., Isgrò, F., Prevete, R., and Tamburrini, G. (2019). Contrastive explanations to classification systems using sparse dictionaries. In *ICIAP*. Springer.

[17] Aranganayagi, S. and Thangavel, K. (2007). Clustering categorical data using silhouette coefficient as a relocating measure. In *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, volume 2, pages 13–17.

[18] Arioua, A. and Croitoru, M. (2015). Formalizing explanatory dialogues. In *International Conference on Scalable Uncertainty Management*, pages 282–297. Springer.

[19] Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

[20] Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010). How to explain individual classification decisions. *JMLR*.

[21] Bakarov, A. (2018). A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*.

[22] Baroni, M., Dinu, G., and Kruszewski, G. (2014). Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL.

[23] Baroni, M., Evert, S., and Lenci, A. (2008). Bridging the gap between semantic theory and computational simulations: Proceedings of the esslli workshop on distributional lexical semantics. *Hamburg, Germany: FOLLI*.

[24] Baroni, M., Murphy, B., Barbu, E., and Poesio, M. (2010). Strudel: A distributional semantic model based on property and types. *Cognitive Science*, 34(2).

[25] Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J. M., and Eckersley, P. (2020). Explainable machine learning in deployment. In *ACM-FAT*.

[26] Biemann, C., Faralli, S., Panchenko, A., and Ponzetto, S. P. (2018). A framework for enriching lexical semantic resources with distributional semantics. *Natural Language Engineering*, 24(2).

[27] Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

[28] Bohus, D. and Rudnicky, A. (2006). A "k hypotheses+ other" belief updating model.

[29] Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *ACL*, 5.

[30] Bordea, G., Buitelaar, P., Faralli, S., and Navigli, R. (2015). Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 902–910.

[31] Bordea, G., Lefever, E., and Buitelaar, P. (2016). Semeval-2016 task 13: Taxonomy extraction evaluation (texeval-2). In *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*, pages 1081–1091.

[32] Boselli, R., Cesarini, M., Marrara, S., Mercorio, F., Mezzanzanica, M., Pasi, G., and Viviani, M. (2018a). Wolmis: a labor market intelligence system for classifying web job vacancies. *Journal of Intelligent Information Systems*, 51(3):477–502.

[33] Boselli, R., Cesarini, M., Mercorio, F., and Mezzanzanica, M. (2013). Inconsistency knowledge discovery for longitudinal data management: A model-based approach. In Holzinger, A. and Pasi, G., editors, *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data - Third International Workshop, HCI-KDD 2013, Held at SouthCHI 2013, Maribor, Slovenia, July 1-3, 2013. Proceedings*, volume 7947 of *Lecture Notes in Computer Science*, pages 183–194. Springer.

[34] Boselli, R., Cesarini, M., Mercorio, F., and Mezzanzanica, M. (2014). A policy-based cleansing and integration framework for labour and healthcare data. In Holzinger, A. and Jurisica, I., editors, *Interactive Knowledge Discovery and Data Mining in Biomedical Informatics - State-of-the-Art and Future Challenges*, volume 8401 of *Lecture Notes in Computer Science*, pages 141–168. Springer.

[35] Boselli, R., Cesarini, M., Mercorio, F., and Mezzanzanica, M. (2017). Using machine learning for labour market intelligence. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 330–342. Springer.

[36] Boselli, R., Cesarini, M., Mercorio, F., and Mezzanzanica, M. (2018b). Classifying online job advertisements through machine learning. *Future Generation Computer Systems*, 86.

[37] Braun, D., Mendez, A. H., Matthes, F., and Langen, M. (2017). Evaluating natural language understanding services for conversational question answering systems. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 174–185.

[38] Bruni, E., Tran, N.-K., and Baroni, M. (2014). Multimodal distributional semantics. *JAIR*, 49.

[39] Bryant, R. E. (1986). Graph-based algorithms for boolean function manipulation. *IEEE Transactions on Computers*.

[40] Bryden, K. (2006). Using a human-in-the-loop evolutionary algorithm to create data-driven music. In *2006 IEEE International Conference on Evolutionary Computation*, pages 2065–2071. IEEE.

[41] Budd, S., Robinson, E. C., and Kainz, B. (2021). A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062.

[42] Bunk, T., Varshneya, D., Vlasov, V., and Nichol, A. (2020). Diet: Lightweight language understanding for dialogue systems. *arXiv preprint arXiv:2004.09936*.

[43] Burkart, N. and Huber, M. F. (2021). A survey on the explainability of supervised machine learning. *JAIR*.

[44] Byrne, R. M. (2019). Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning. In *IJCAI*, pages 6276–6282.

[45] Camacho-Collados, J. and Pilehvar, M. T. (2018). From word to sense embeddings: A survey on vector representations of meaning. *JAIR*, 63.

[46] Camacho-Collados, J., Pilehvar, M. T., Collier, N., and Navigli, R. (2017). Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26.

[47] Cambria, E., Hussain, A., Havasi, C., and Eckl, C. (2009). Common sense computing: From the society of mind to digital intuition and beyond. In Fierrez, J., Ortega, J., Esposito, A., Drygajlo, A., and Faundez-Zanuy, M., editors, *Biometric ID Management and Multimodal Communication*, volume 5707 of *Lecture Notes in Computer Science*, pages 252–259. Springer, Berlin Heidelberg.

[48] Cambria, E., Liu, Q., Decherchi, S., Xing, F., , and Kwok, K. (2022). SenticNet 7: A commonsense-based neurosymbolic ai framework for explainable sentiment analysis. In *LREC*.

[49] Cambria, E., Schuller, B., Xia, Y., and White, B. (2016). New avenues in knowledge bases for natural language processing. *Knowledge-Based Systems*, 108:1–4.

[50] Caruana, R., Lundberg, S., Ribeiro, M. T., Nori, H., and Jenkins, S. (2020). Intelligible and explainable machine learning: Best practices and practical challenges. In *ACM-SIGKDD*, pages 3511–3512.

[51] Caselles-Dupré, H., Lesaint, F., and Royo-Letelier, J. (2018). Word2vec applied to recommendation: Hyperparameters matter. In *RECSYS*.

[52] CEDEFOP (2016). Real-time labour market information on skill requirements: Setting up the eu system for online vacancy analysis. https://goo.gl/5FZS3E.

[53] Chang, S., Harper, F. M., and Terveen, L. G. (2016). Crowd-based personalized natural language explanations for recommendations. In *ACM Conference on Recommender Systems*.

[54] Chaves, A. P. and Gerosa, M. A. (2020). How should my chatbot interact? a survey on social characteristics in human–chatbot interaction design. *International Journal of Human–Computer Interaction*, pages 1–30.

[55] Chen, D. L. and Mooney, R. J. (2008). Learning to sportscast: a test of grounded language acquisition. In *ICML*.

[56] Chen, Q., Zhuo, Z., and Wang, W. (2019). Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

[57] Chen, Y.-N., Hakanni-Tür, D., Tur, G., Celikyilmaz, A., Guo, J., and Deng, L. (2016). Syntax or semantics? knowledge-guided joint semantic frame parsing. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 348–355. IEEE.

[58] Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

[59] Chromik, M. and Butz, A. (2021). Human-xai interaction: A review and design principles for explanation user interfaces. In *IFIP Conference on Human-Computer Interaction*, pages 619–640. Springer.

[60] Ciatto, G., Schumacher, M. I., Omicini, A., and Calvaresi, D. (2020). Agent-based explanations in ai: towards an abstract framework. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 3–20. Springer.

[61] Cohen, J. (1992). A power primer. *Psychological bulletin*, 112(1).

[62] Colace, F., De Santo, M., Lombardi, M., Mercorio, F., Mezzanzanica, M., and Pascale, F. (2019). Towards labour market intelligence through topic modelling. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*.

[63] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*. ACM.

[64] Colombo, E., Mercorio, F., and Mezzanzanica, M. (2019). AI meets labor market: exploring the link between automation and skills. *Information Economics and Policy*, 47.

[65] Core, M. G., Lane, H. C., Van Lent, M., Gomboc, D., Solomon, S., and Rosenberg, M. (2006). Building explainable artificial intelligence systems. In *AAAI*.

[66] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3).

[67] Costa, F., Ouyang, S., Dolog, P., and Lawlor, A. (2018). Automatic generation of natural language explanations. In *IUI*.

[68] Čyras, K., Rago, A., Albini, E., Baroni, P., and Toni, F. (2021). Argumentative xai: a survey. In *IJCAI*, page 4392–4399.

[69] Dazeley, R., Vamplew, P., Foale, C., Young, C., Aryal, S., and Cruz, F. (2021). Levels of explainable artificial intelligence for human-aligned conversational explanations. *Artificial Intelligence*, 299:103525.

[70] De Gennaro, M., Krumhuber, E. G., and Lucas, G. (2020). Effectiveness of an empathic chatbot in combating adverse effects of social exclusion on mood. *Frontiers in psychology*, 10:3061.

[71] Deeks, A. (2019). The judicial demand for explainable artificial intelligence. *Columbia Law Review*, 119(7):1829–1850.

[72] Dennett, D. C. (1989). *The intentional stance*. MIT press.

[73] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

[74] Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *ACL*.

[75] Di Cicco, M., Potena, C., Grisetti, G., and Pretto, A. (2017). Automatic model based dataset generation for fast and accurate crop and weeds detection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5188–5195. IEEE.

[76] Divya, S., Indumathi, V., Ishwarya, S., Priyasankari, M., and Devi, S. K. (2018). A self-diagnosis medical chatbot using artificial intelligence. *Journal of Web Development and Web Designing*, 3(1):1–7.

[77] Donadello, I. and Dragoni, M. (2021). Bridging signals to natural language explanations with explanation graphs.

[78] Doran, D., Schulz, S., and Besold, T. R. (2017). What does explainable AI really mean? A new conceptualization of perspectives. In Besold, T. R. and Kutz, O., editors, *AI*IA*.

[79] Ehsan, U., Tambwekar, P., Chan, L., Harrison, B., and Riedl, M. O. (2019). Automated rationale generation: a technique for explainable ai and its effects on human perceptions. In *IUI*.

[80] EuroStat (2020). Towards the european web intelligence hub — european system for collection and analysis of online job advertisement data (wih-oja), available at https://tinyurl.com/y3xqzfhp.

[81] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9(Aug).

[82] Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.

[83] Fei-Fei, L., Fergus, R., and Perona, P. (2006). One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611.

[84] Figueira, J., Greco, S., and Ehrgott, M. (2005). *Multiple criteria decision analysis: state of the art surveys*.

[85] Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. In *WWW*.

[86] Frey, C. B. and Osborne, M. A. (2017). The future of employment: How susceptible are jobs to computerisation? *Technological Forecasting and Social Change*, 114(Supplement C).

[87] Fu, R., Guo, J., Qin, B., Che, W., Wang, H., and Liu, T. (2014). Learning semantic hierarchies via word embeddings. In *ACL*.

[88] Gao, X., Gong, R., Zhao, Y., Wang, S., Shu, T., and Zhu, S.-C. (2020). Joint mind modeling for explanation generation in complex human-robot collaborative tasks. In *2020 29th IEEE international conference on robot and human interactive communication (RO-MAN)*, pages 1119–1126. IEEE.

[89] Gatt, A. and Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *JAIR*, 61.

[90] Gedikli, F., Jannach, D., and Ge, M. (2014). How should i explain? a comparison of different explanation types for recommender systems. *International Journal of Human-Computer Studies*, 72(4).

[91] Ghannay, S., Favre, B., Esteve, Y., and Camelin, N. (2016). Word embedding evaluation and combination. In *LREC*.

[92] Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., and Seveso, A. (2020a). Neo: A tool for taxonomy enrichment with new emerging occupations. In *ISWC*, pages 568–584.

[93] Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., and Seveso, A. (2020b). NEO: A tool for taxonomy enrichment with new emerging occupations. In *The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference*, volume 12507 of *Lecture Notes in Computer Science*, pages 568–584. Springer.

[94] Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., and Seveso, A. (2021a). NEO: A system for identifying new emerging occupation from job ads. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pages 16035–16037. AAAI Press.

[95] Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., and Seveso, A. (2021b). Neo: A system for identifying new emerging occupation from job ads. In *The 35th AAAI Conference on Artificial Intelligence - Demo Track*.

[96] Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., and Seveso, A. (2021c). Neo: A system for identifying new emerging occupation from job ads. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 16035–16037.

[97] Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., and Seveso, A. (2021d). Skills2graph: Processing million job ads to face the job skill mismatch problem. In Zhou, Z., editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 4984–4987. ijcai.org.

[98] Giabelli, A., Malandri, L., Mercorio, F., Mezzanzanica, M., and Seveso, A. (2021e). Skills2job: A recommender system that encodes job offer embeddings on graph databases. *Appl. Soft Comput.*, 101:107049.

[99] Gkatzia, D., Lemon, O., and Rieser, V. (2016). Natural language generation enhances human decision-making with uncertain information. In *ACL*.

[100] Gladkova, A., Drozd, A., and Matsuoka, S. (2016). Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *NAACL*.

[101] Glaser, B. G., Strauss, A. L., and Strutzel, E. (1968). The discovery of grounded theory; strategies for qualitative research. *Nursing research*, 17(4):364.

[102] Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *J COMPUT GRAPH STAT*.

[103] Goo, C.-W., Gao, G., Hsu, Y.-K., Huo, C.-L., Chen, T.-C., Hsu, K.-W., and Chen, Y.-N. (2018). Slot-gated modeling for joint slot filling and intent prediction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 753–757.

[104] Grinberg, M. (2018). *Flask web development: developing web applications with python*. " O'Reilly Media, Inc.".

[105] Guidotti, R., Monreale, A., Ruggieri, S., Pedreschi, D., Turini, F., and Giannotti, F. (2018a). Local rule-based explanations of black box decision systems. *arXiv preprint arXiv:1805.10820*.

[106] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., and Pedreschi, D. (2018b). A survey of methods for explaining black box models. *CSUR*, 51(5).

[107] Guo, D., Tur, G., Yih, W.-t., and Zweig, G. (2014). Joint semantic utterance classification and slot filling with recursive neural networks. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 554–559. IEEE.

[108] Gupta, A., Zhang, P., Lalwani, G., and Diab, M. (2019). Casa-nlu: Context-aware self-attentive natural language understanding for task-oriented chatbots. *arXiv preprint arXiv:1909.08705*.

[109] Halawi, G., Dror, G., Gabrilovich, E., and Koren, Y. (2012). Large-scale learning of word relatedness with constraints. In *ACM SIGKDD*.

[110] Hall, M., Harborne, D., Tomsett, R., Galetic, V., Quintana-Amate, S., Nottle, A., and Preece, A. (2019). A systematic method to understand requirements for explainable ai (xai) systems. In *IJCAI*.

[111] Hall, P., Gill, N., Kurka, M., and Phan, W. (2017). Machine learning interpretability with h2o driverless ai. *H2O. ai. URL: http://docs. h2o. ai/driverless-ai/latest-stable/docs/booklets/MLIBooklet. pdf*.

[112] Halpern, J. Y. and Pearl, J. (2005). Causes and explanations: A structural-model approach. part i: Causes. *The British journal for the philosophy of science*, 56(4).

[113] Heckerman, D. and Horvitz, E. (1998). Inferring informational goals from free-text queries: a bayesian approach. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 230–237.

[114] Hendricks, L. A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., and Darrell, T. (2016). Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer.

[115] Hendricks, L. A., Hu, R., Darrell, T., and Akata, Z. (2018a). Generating counterfactual explanations with natural language. In *ICML WHI*.

[116] Hendricks, L. A., Hu, R., Darrell, T., and Akata, Z. (2018b). Grounding visual explanations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 264–279.

[117] Henelius, A., Puolamäki, K., and Ukkonen, A. (2017). Interpreting classifiers through attribute interactions in datasets. In *ICML WHI*.

[118] Hernandez-Bocanegra, D. C. and Ziegler, J. (2021). Conversational review-based explanations for recommender systems: Exploring users' query behavior. In *CUI 2021-3rd Conference on Conversational User Interfaces*, pages 1–11.

[119] Hill, F., Reichart, R., and Korhonen, A. (2015). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4).

[120] Hilton, D. J. (1990). Conversational processes and causal explanation. *Psychological Bulletin*, 107(1).

[121] Hoffman, R. R., Klein, G., and Mueller, S. T. (2018a). Explaining explanation for "explainable ai". In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 62, pages 197–201. SAGE Publications Sage CA: Los Angeles, CA.

[122] Hoffman, R. R., Mueller, S. T., Klein, G., and Litman, J. (2018b). Metrics for explainable ai: Challenges and prospects. *arXiv preprint arXiv:1812.04608*.

[123] Hohman, F., Srinivasan, A., and Drucker, S. M. (2019). Telegam: Combining visualization and verbalization for interpretable machine learning. In *2019 IEEE Visualization Conference (VIS)*. IEEE.

[124] Holzinger, A. (2016). Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain Informatics*, 3(2):119–131.

[125] Holzinger, A., Carrington, A., and Müller, H. (2020). Measuring the quality of explanations: the system causability scale (scs). *KI-Künstliche Intelligenz*, 34(2):193–198.

[126] Holzinger, A., Langs, G., Denk, H., Zatloukal, K., and Müller, H. (2019). Causability and explainability of artificial intelligence in medicine. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(4):e1312.

[127] Holzinger, A., Malle, B., Saranti, A., and Pfeifer, B. (2021). Towards multi-modal causability with graph neural networks enabling information fusion for explainable ai. *Information Fusion*, 71:28–37.

[128] Horvitz, E. and Paek, T. (1999). A computational architecture for conversation. In *UM99 User Modeling*, pages 201–210. Springer.

[129] Hovorka, D. S., Germonprez, M., and Larsen, K. R. (2008). Explanation in information systems. *ISJ*.

[130] Hua, W., Wang, Z., Wang, H., Zheng, K., and Zhou, X. (2016). Understand short texts by harvesting and analyzing semantic knowledge. *IEEE transactions on Knowledge and data Engineering*, 29(3).

[131] Huysmans, J., Dejaeger, K., Mues, C., Vanthienen, J., and Baesens, B. (2011). An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154.

[132] Jentzsch, S. F., Höhn, S., and Hochgeschwender, N. (2019). Conversational interfaces for explainable ai: a human-centred approach. In *International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems*, pages 77–92. Springer.

[133] Jeong, M. and Lee, G. G. (2008). Triangular-chain conditional random fields. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(7):1287–1302.

[134] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

[135] Jin, P., Zhang, Y., Chen, X., and Xia, Y. (2016). Bag-of-embeddings for text classification. In *IJCAI*, pages 2824–2830.

[136] Jing, B., Xie, P., and Xing, E. (2018). On the automatic generation of medical imaging reports. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2577–2586.

[137] Johansson, U., Niklasson, L., and König, R. (2004). Accuracy vs. comprehensibility in data mining models. In *Proceedings of the seventh international conference on information fusion*, volume 1, pages 295–300. Citeseer.

[138] Johs, A. J., Agosto, D. E., and Weber, R. O. (2020). Qualitative investigation in explainable artificial intelligence: A bit more insight from social science. *arXiv preprint arXiv:2011.07130*.

[139] Jurafsky, D. and Martin, J. H. (2020). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*.

[140] Karalus, J., Halilovic, A., and Lindner, F. (2021). Explanations in, explanations out: human-in-the-loop social navigation learning. In *ICDL Workshop on Human aligned Reinforcement Learning for Autonomous Agents and Robots*.

[141] Kass, A. and Leake, D. (1987). Types of explanations. Technical report, YALE UNIV NEW HAVEN CT DEPT OF COMPUTER SCIENCE.

[142] Kato, H. and Harada, T. (2014). Image reconstruction from bag-of-visual-words. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 955–962.

[143] Kenny, E. M., Ford, C., Quinn, M., and Keane, M. T. (2021). Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. *Artificial Intelligence*, 294:103459.

[144] Keselj, V. (2009). Speech and language processing daniel jurafsky and james h. martin (stanford university and university of colorado at boulder) pearson prentice hall, 2009, xxxi+ 988 pp; hardbound, isbn 978-0-13-187321-6.

[145] Kiela, D., Hill, F., and Clark, S. (2015). Specializing word embeddings for similarity or relatedness. In *EMNLP*.

[146] Kim, B., Glassman, E., Johnson, B., and Shah, J. (2015a). ibcm: Interactive bayesian case model empowering humans via intuitive interaction.

[147] Kim, B., Shah, J., and Doshi-Velez, F. (2015b). Mind the gap: a generative approach to interpretable feature selection and extraction. In *NIPS*.

[148] Kitzelmann, E., Schmid, U., Olsson, R., and Kaelbling, L. P. (2006). Inductive synthesis of functional programs: An explanation based generalization approach. *Journal of Machine Learning Research*, 7(2).

[149] Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *ACL*.

[150] Köhn, A. (2015). What's in an embedding? analyzing word embeddings through multilingual evaluation. In *EMNLP*.

[151] Korpan, R. and Epstein, S. L. (2018). Toward natural explanations for a robot's navigation plans. *HRI*.

[152] Krahmer, E. and Theune, M. (2010). *Empirical methods in natural language generation: Data-oriented methods and empirical evaluation*, volume 5790. Springer.

[153] Krippendorff, K. (2004). Reliability in content analysis: Some common misconceptions and recommendations. *Human communication research*, 30(3):411–433.

[154] Kulesza, T., Stumpf, S., Burnett, M., and Kwan, I. (2012). Tell me more? the effects of mental model soundness on personalizing an intelligent agent. In *SIGCHI*.

[155] Kuzba, M. (2020). What would you ask the machine learning model? identification of user needs for model explanations based on human-model conversations. In *ECML PKDD 2020 Workshops*, volume 1323, page 447. Springer Nature.

[156] Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[157] Langer, M., Oster, D., Speith, T., Hermanns, H., Kästner, L., Schmidt, E., Sesing, A., and Baum, K. (2021). What do we want from explainable artificial intelligence (xai)?–a stakeholder perspective on xai and a conceptual model guiding interdisciplinary xai research. *Artificial Intelligence*, 296:103473.

[158] Lastra-Díaz, J. J., García-Serrano, A., Batet, M., Fernández, M., and Chirigati, F. (2017). Hesml: A scalable ontology-based semantic similarity measures library with a set of reproducible experiments and a replication dataset. *Information Systems*, 66.

[159] Lau, J. H. and Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

[160] Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2).

[161] Lecue, F. (2020). On the role of knowledge graphs in explainable ai. *Semantic Web*, 11(1):41–51.

[162] Lee, S. (2013). Structured discriminative model for dialog state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 442–451.

[163] Lei, T., Barzilay, R., and Jaakkola, T. (2016). Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

[164] Letham, B., Rudin, C., McCormick, T. H., Madigan, D., et al. (2015). Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3).

[165] Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *NeurIPS*.

[166] Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3.

[167] Li, W., Shao, W., Ji, S., and Cambria, E. (2022). Bieru: Bidirectional emotional recurrent unit for conversational sentiment analysis. *Neurocomputing*, 467:73–82.

[168] Li, Y., Pan, Q., Wang, S., Yang, T., and Cambria, E. (2018). A generative model for category text generation. *Information Sciences*, 450:301–315.

[169] Liao, Q. V., Gruen, D., and Miller, S. (2020). Questioning the ai: informing design practices for explainable ai user experiences. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

[170] Lin, D. et al. (1998). An information-theoretic definition of similarity. In *ICML*, volume 98.

[171] Lipton, P. (1990). Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27:247–266.

[172] Liu, B. and Lane, I. (2016). Attention-based recurrent neural network models for joint intent detection and slot filling. *arXiv preprint arXiv:1609.01454*.

[173] Liu, P., Zhang, L., and Gulla, J. A. (2020). Dynamic attention-based explainable recommendation with textual and visual fusion. *Information Processing & Management*, 57(6):102099.

[174] Liu, X., Eshghi, A., Swietojanski, P., and Rieser, V. (2019). Benchmarking natural language understanding services for building conversational agents. In *10th International Workshop on Spoken Dialogue Systems Technology 2019*.

[175] Liza, F. F. and Grzes, M. (2016). An improved crowdsourcing based evaluation technique for word embedding methods. In *Workshop on Evaluating Vector-Space Representations for NLP*.

[176] Lorenc, P., Marek, P., Pichl, J., Konrád, J., and Šedivỳ, J. (2020). Do we need online nlu tools? *arXiv preprint arXiv:2011.09825*.

[177] Lou, Y., Caruana, R., and Gehrke, J. (2012). Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158.

[178] Lucic, A., Haned, H., and de Rijke, M. (2020). Why does my model fail? contrastive local explanations for retail forecasting. In *ACM FAccT*.

[179] Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems*.

[180] Ma, Y., Nguyen, K. L., Xing, F., and Cambria, E. (2020). A survey on empathetic dialogue systems. *Information Fusion*, 64:50–70.

[181] Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *ACL HLT*.

[182] Madumal, P., Miller, T., Sonenberg, L., and Vetere, F. (2019). A grounded interaction protocol for explainable artificial intelligence. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1033–1041.

[183] Maedche, A. and Staab, S. (2001). Ontology learning for the semantic web. *IEEE Intelligent systems*, 16(2).

[184] Maedche, A. and Volz, R. (2001). The ontology extraction & maintenance framework text-to-onto. In *Proc. Workshop on Integrating Data Mining and Knowledge Management, USA*.

[185] Malandri, L., Mercorio, F., Mezzanzanica, M., and Nobani, N. (2020). Meet: A method for embeddings evaluation for taxonomic data. In *2020 International Conference on Data Mining Workshops (ICDMW)*, pages 31–38. IEEE.

[186] Malandri, L., Mercorio, F., Mezzanzanica, M., and Nobani, N. (2021a). MEET-LM: A method for embeddings evaluation for taxonomic data in the labour market. *Comput. Ind.*, 124:103341.

[187] Malandri, L., Mercorio, F., Mezzanzanica, M., and Nobani, N. (2021b). Taxoref: Embeddings evaluation for ai-driven taxonomy refinement. In Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., and Lozano, J. A., editors, *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference, ECML PKDD 2021, Bilbao, Spain, September 13-17, 2021, Proceedings, Part III*, volume 12977 of *Lecture Notes in Computer Science*, pages 612–627. Springer.

[188] Malandri, L., Mercorio, F., Mezzanzanica, M., Nobani, N., and Seveso, A. (2022a). ContrXT: Generating contrastive explanations from any text classifier. *Information Fusion, special issue on XAI*, 81:103–115.

[189] Malandri, L., Mercorio, F., Mezzanzanica, M., Nobani, N., and Seveso, A. (2022b). ContrXT PyPI project page. https://pypi.org/project/contrxt/. Accessed: 2022-05-20.

[190] Malandri, L., Mercorio, F., Mezzanzanica, M., Nobani, N., and Seveso, A. (2022c). ContrXT web page. https://ContrXT.ai. Accessed: 2022-05-20.

[191] Malone, T., Creedon, M., and Malone, J. (1970). Human factors engineering for maritime systems. *WIT Transactions on The Built Environment*, 39.

[192] Mao, J., Gan, C., Kohli, P., Tenenbaum, J. B., and Wu, J. (2018). The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. In *International Conference on Learning Representations*.

[193] Mariotti, E., Alonso, J. M., and Gatt, A. (2020). Towards harnessing natural language generation to explain black-box models. In *NL4XAI*.

[194] Martens, D., Baesens, B., Van Gestel, T., and Vanthienen, J. (2007). Comprehensible credit scoring models using rule extraction from support vector machines. *European journal of operational research*, 183(3):1466–1476.

[195] McBurney, P. and Parsons, S. (2002). Games that agents play: A formal framework for dialogues between autonomous agents. *Journal of logic, language and information*, 11(3):315–334.

[196] McClure, J. (2002). Goal-based explanations of actions and outcomes. *European review of social psychology*, 12(1):201–235.

[197] McGill, A. L. and Klein, J. G. (1993). Contrastive and counterfactual reasoning in causal judgment. *Journal of Personality and Social Psychology*, 64(6).

[198] Meng, Y., Huang, J., Wang, G., Zhang, C., Zhuang, H., Kaplan, L., and Han, J. (2019). Spherical text embedding. In *Advances in Neural Information Processing Systems*, pages 8208–8217.

[199] Mezzanzanica, M., Boselli, R., Cesarini, M., and Mercorio, F. (2012). Data quality sensitivity analysis on aggregate indicators. In Helfert, M., Francalanci, C., and Filipe, J., editors, *DATA 2012 - Proceedings of the International Conference on Data Technologies and Applications, Rome, Italy, 25-27 July, 2012*, pages 97–108. SciTePress.

[200] Mezzanzanica, M., Boselli, R., Cesarini, M., and Mercorio, F. (2015). A model-based evaluation of data quality activities in KDD. *Inf. Process. Manag.*, 51(2):144–166.

[201] Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[202] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*.

[203] Miller, G. A. and Charles, W. G. (1991). Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1).

[204] Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*.

[205] Miller, T. (2021). Contrastive explanation: A structural-model approach. *The Knowledge Engineering Review*, 36.

[206] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., and Gao, J. (2021). Deep learning based text classification: A comprehensive review. *ACM Computing Surveys*, 54(3):Article 62.

[207] Mittelstadt, B., Russell, C., and Wachter, S. (2019). Explaining explanations in ai. In *ACM FAccT*.

[208] Muller, H., Mayrhofer, M. T., Van Veen, E.-B., and Holzinger, A. (2021). The ten commandments of ethical medical ai. *Computer*, 54(07):119–123.

[209] Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (2015). Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.

[210] Nguyen, K. A., Köper, M., Walde, S. S. i., and Vu, N. T. (2017). Hierarchical embeddings for hypernymy detection and directionality. *arXiv preprint arXiv:1707.07273*.

[211] Ni, J., Pandelea, V., Young, T., Zhou, H., and Cambria, E. (2022). Hitkg: Towards goal-oriented conversations via multi-hierarchy learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 11112–11120.

[212] O'Hara, T. D., Hugall, A. F., Thuy, B., Stöhr, S., and Martynov, A. V. (2017). Restructuring higher taxonomy using broad-scale phylogenomics: the living ophiuroidea. *Molecular phylogenetics and evolution*, 107:415–430.

[213] Olden, J. D. and Jackson, D. A. (2002). Illuminating the "black box": a randomization approach for understanding variable contributions in artificial neural networks. *Ecological modelling*, 154(1-2):135–150.

[214] Ortega, A., Fierrez, J., Morales, A., Wang, Z., and Ribeiro, T. (2021). Symbolic ai for xai: Evaluating lfit inductive programming for fair and explainable automatic recruitment. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 78–87.

[215] Palan, S. and Schitter, C. (2018). Prolific. ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27.

[216] Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[217] Papamichail, K. N. and French, S. (2003). Explaining and justifying the advice of a decision support system: a natural language generation approach. *Expert Systems with Applications*, 24(1).

[218] Park, D. H., Hendricks, L. A., Akata, Z., Rohrbach, A., Schiele, B., Darrell, T., and Rohrbach, M. (2018). Multimodal explanations: Justifying decisions and pointing to the evidence. In *CVPR*.

[219] Payrovnaziri, S. N., Chen, Z., Rengifo-Moreno, P., Miller, T., Bian, J., Chen, J. H., Liu, X., and He, Z. (2020). Explainable artificial intelligence models using real-world electronic health record data: a systematic scoping review. *Journal of the American Medical Informatics Association*, 27(7):1173–1185.

[220] Pedelty, M. (1965). A review of the field of artificial intelligence and its possible applications to nasa objectives final report.

[221] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12.

[222] Peng, B., Li, X., Gao, J., Liu, J., and Wong, K.-F. (2018). Deep dyna-q: Integrating planning for task-completion dialogue policy learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2182–2192.

[223] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *EMNLP*.

[224] Perone, C. S., Silveira, R., and Paula, T. S. (2018). Evaluation of sentence embeddings in downstream and linguistic probing tasks. *arXiv preprint arXiv:1806.06259*.

[225] Ponzetto, S. P. and Navigli, R. (2009). Large-scale taxonomy mapping for restructuring and integrating wikipedia. In *IJCAI*, volume 9, pages 2083–2088.

[226] Poulin, B., Eisner, R., Szafron, D., Lu, P., Greiner, R., Wishart, D. S., Fyshe, A., Pearcy, B., MacDonell, C., and Anvik, J. (2006). Visual explanation of evidence with additive classifiers. In *Proceedings Of The National Conference On Artificial Intelligence*, volume 21, page 1822. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

[227] Powell, J., Sentz, K., and Klein, M. (2021). Human-in-the-loop refinement of word embeddings. *arXiv preprint arXiv:2110.02884*.

[228] Press, O. and Wolf, L. (2016). Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*.

[229] Przybyła, P. and Soto, A. J. (2021). When classification accuracy is not enough: Explaining news credibility assessment. *Information Processing & Management*, 58(5):102653.

[230] Qian, K., Danilevsky, M., Katsis, Y., Kawas, B., Oduor, E., Popa, L., and Li, Y. (2021). Xnlp: A living survey for xai research in natural language processing. In *26th International Conference on Intelligent User Interfaces*, pages 78–80.

[231] Radinsky, K., Agichtein, E., Gabrilovich, E., and Markovitch, S. (2011). A word at a time: computing word relatedness using temporal semantic analysis. In *WWW*.

[232] Rajaraman, A. and Ullman, J. D. (2011). *Mining of massive datasets*. Cambridge University Press.

[233] Raman, V., Lignos, C., Finucane, C., Lee, K. C., Marcus, M. P., and Kress-Gazit, H. (2013). Sorry dave, i'm afraid i can't do that: Explaining unachievable robot tasks using natural language. In *Robotics: Science and Systems*.

[234] Raymond, A., Gunes, H., and Prorok, A. (2020). Culture-based explainable human-agent deconfliction. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1107–1115.

[235] Reiter, E. and Dale, R. (1997). Building applied natural language generation. *Natural Language Engineering*, 3(1).

[236] Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Studies in Natural Language Processing.

[237] Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *JAIR*, 11.

[238] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). Why should i trust you?: Explaining the predictions of any classifier. In *ACM-SIGKDD*, pages 1135–1144.

[239] Ribeiro, M. T., Singh, S., and Guestrin, C. (2018). Anchors: High-precision model-agnostic explanations. In *AAAI*, volume 18, pages 1527–1535.

[240] Ribera, M. and Lapedriza, A. (2019). Can we do better explanations? a proposal of user-centered explainable ai. In *IUI Workshops*.

[241] Robeer, M. J. (2018). Contrastive explanation for machine learning. Master's thesis.

[242] Roller, S., Dinan, E., Goyal, N., Ju, D., Williamson, M., Liu, Y., Xu, J., Ott, M., Smith, E. M., Boureau, Y.-L., et al. (2021). Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325.

[243] Rosenthal, S., Selvaraj, S. P., and Veloso, M. M. (2016). Verbalization: Narration of autonomous robot experience. In *IJCAI*, volume 16, pages 862–868.

[244] Rubenstein, H. and Goodenough, J. B. (1965). Contextual correlates of synonymy. *Communications of the ACM*, 8(10).

[245] Saaty, R. W. (1987). The analytic hierarchy process—what it is and how it is used. *Mathematical modelling*, 9(3-5):161–176.

[246] Saaty, T. L. (2004). Fundamentals of the analytic network process—multiple networks with benefits, costs, opportunities and risks. *JSSI*.

[247] Schnabel, T., Labutov, I., Mimno, D., and Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *EMNLP*.

[248] Schönbrodt, F. D. and Perugini, M. (2013). At what sample size do correlations stabilize? *Journal of Research in Personality*, 47(5).

[249] Schoonderwoerd, T. A., Jorritsma, W., Neerincx, M. A., and Van Den Bosch, K. (2021). Human-centered xai: Developing design patterns for explanations of clinical decision support systems. *International Journal of Human-Computer Studies*, 154:102684.

[250] Sebastiani, F. (2002). Machine learning in automated text categorization. *CSUR*, 34(1).

[251] Seco, N., Veale, T., and Hayes, J. (2004). An intrinsic information content metric for semantic similarity in wordnet. In *Ecai*, volume 16.

[252] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

[253] Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., and Pineau, J. (2015). Hierarchical neural network generative models for movie dialogues. *arXiv preprint arXiv:1507.04808*, 7(8):434–441.

[254] Settles, B. (2009). Active learning literature survey.

[255] Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

[256] Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Deep inside convolutional networks: Visualising image classification models and saliency maps. In *In Workshop at International Conference on Learning Representations*. Citeseer.

[257] Sklar, E. I. and Azhar, M. Q. (2018). Explanation through argumentation. In *Proceedings of the 6th International Conference on Human-Agent Interaction*, pages 277–285.

[258] Sokol, K. and Flach, P. (2020a). Limetree: Interactively customisable explanations based on local surrogate multi-output regression trees. *arXiv preprint arXiv:2005.01427*.

[259] Sokol, K. and Flach, P. (2020b). One explanation does not fit all. *KI-Künstliche Intelligenz*, pages 1–16.

[260] Sokol, K. and Flach, P. A. (2018a). Conversational explanations of machine learning predictions through class-contrastive counterfactual statements. In *IJCAI*.

[261] Sokol, K. and Flach, P. A. (2018b). Glass-box: Explaining AI decisions with counterfactual statements through conversation with a voice-enabled virtual assistant. In *IJCAI*, pages 5868–5870.

[262] Sreedharan, S., Srivastava, S., and Kambhampati, S. (2021). Using state abstractions to compute personalized contrastive explanations for ai agent behavior. *Artificial Intelligence*, 301:103570.

[263] Sripada, S., Reiter, E., and Davy, I. (2003). Sumtime-mousam: Configurable marine weather forecast generator. *Expert Update*, 6(3).

[264] Stöger, K., Schneeberger, D., and Holzinger, A. (2021). Medical artificial intelligence: the european legal perspective. *Communications of the ACM*, 64(11):34–36.

[265] Sturm, I., Lapuschkin, S., Samek, W., and Müller, K.-R. (2016). Interpretable deep neural networks for single-trial eeg classification. *Journal of neuroscience methods*, 274:141–145.

[266] Sun, K., Chen, L., Zhu, S., and Yu, K. (2014). A generalized rule based tracker for dialogue state tracking. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 330–335. IEEE.

[267] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

[268] Szafron, D., Poulin, B., Eisner, R., Lu, P., Greiner, R., Wishart, D., Fyshe, A., Pearcy, B., Macdonell, C., and Anvik, J. (2006). Visual explanation of evidence in additive classifiers. In *Proceedings of innovative applications of artificial intelligence*, volume 2.

[269] Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*.

[270] Tiwari, A., Saha, T., Saha, S., Sengupta, S., Maitra, A., Ramnani, R., and Bhattacharyya, P. (2021). Multi-modal dialogue policy learning for dynamic and co-operative goal setting. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

[271] Tomsett, R., Braines, D., Harborne, D., Preece, A., and Chakraborty, S. (2018). Interpretable to whom? a role-based model for analyzing interpretable machine learning systems. In *ICML*.

[272] Tsiakas, K. and Murray-Rust, D. (2022). Using human-in-the-loop and explainable ai to envisage new future work practices. In *Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 588–594.

[273] Tsvetkov, Y., Faruqui, M., Ling, W., Lample, G., and Dyer, C. (2015). Evaluation of word vector representations by subspace alignment. In *EMNLP*.

[274] Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *ACL*.

[275] Turner, R. (2016). A model explanation system. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE.

[276] Van Bouwel, J. and Weber, E. (2002). Remote causes, bad explanations? *JTSB*, 32(4).

[277] van der Waa, J., Robeer, M., van Diggelen, J., Brinkhuis, M., and Neerincx, M. (2018). Contrastive explanations with local foil trees. In *Proceedings of the ICML Workshop on Human Interpretability in Machine Learning (WHI 2018), Stockholm, Sweden*, volume 37.

[278] Vanzo, A., Bastianelli, E., and Lemon, O. (2019). Hierarchical multi-task natural language understanding for cross-domain conversational ai: Hermit nlu. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 254–263.

[279] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

[280] Vedula, N., Nicholson, P. K., Ajwani, D., Dutta, S., Sala, A., and Parthasarathy, S. (2018). Enriching taxonomies with functional domain knowledge. In *ACM SIGIR*.

[281] Verbeke, W., Martens, D., Mues, C., and Baesens, B. (2011). Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert systems with applications*, 38(3):2354–2364.

[282] Verma, S., Dickerson, J., and Hines, K. (2020). Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596*.

[283] Vilone, G. and Longo, L. (2020). Explainable artificial intelligence: a systematic review. *arXiv preprint arXiv:2006.00093*.

[284] Vilone, G. and Longo, L. (2021). Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion*, 76:89–106.

[285] Vinel, M., Ryazanov, I., Botov, D., and Nikolaev, I. (2019). Experimental comparison of unsupervised approaches in the task of separating specializations within professions in job vacancies. In *Conference on Artificial Intelligence and Natural Language*. Springer.

[286] Wachter, S., Mittelstadt, B., and Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841.

[287] Walton, D. (2007). Dialogical models of explanation. *ExaCt*, 2007:1–9.

[288] Walton, D. (2011). A dialogue system specification for explanation. *Synthese*, 182(3):349–374.

[289] Walton, D. N. and Krabbe, E. C. (1995). *Commitment in dialogue: Basic concepts of interpersonal reasoning*. SUNY press.

[290] Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C.-C. J. (2019a). Evaluating word embedding models: methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, 8.

[291] Wang, C., He, X., and Zhou, A. (2017). A short survey on taxonomy learning from text corpora: Issues, resources and recent advances. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1190–1203.

[292] Wang, D., Yang, Q., Abdul, A., and Lim, B. Y. (2019b). Designing theory-driven user-centric explainable ai. In *CHI*.

[293] Wang, R. Y. and Strong, D. M. (1996). Beyond accuracy: What data quality means to data consumers. *Journal of management information systems*, 12(4).

[294] Wang, Y., Shen, Y., and Jin, H. (2018). A bi-model based rnn semantic frame parsing model for intent detection and slot filling. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 309–314.

[295] Wang, Z. and Lemon, O. (2013). A simple and generic belief tracking mechanism for the dialog state tracking challenge: On the believability of observed information. In *Proceedings of the SIGDIAL 2013 Conference*, pages 423–432.

[296] Webber, B., Egg, M., and Kordoni, V. (2012). Discourse structure and language technology. *Natural Language Engineering*, 18(4):437.

[297] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, 3(1):1–40.

[298] Weld, H., Huang, X., Long, S., Poon, J., and Han, S. C. (2021). A survey of joint intent detection and slot-filling models in natural language understanding. *arXiv preprint arXiv:2101.08091*.

[299] Werner, C. (2020). Explainable ai through rule-based interactive conversation. In *EDBT/ICDT*.

[300] Williams, J. D., Raux, A., and Henderson, M. (2016). The dialog state tracking challenge series: A review. *Dialogue & Discourse*, 7(3):4–33.

[301] Williams, J. D. and Zweig, G. (2016). End-to-end lstm-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*.

[302] Wu, W., Li, H., Wang, H., and Zhu, K. Q. (2012). Probase: A probabilistic taxonomy for text understanding. In *ACM SIGMOD*.

[303] Wu, X., Xiao, L., Sun, Y., Zhang, J., Ma, T., and He, L. (2022). A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*.

[304] Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *ACL*.

[305] Xia, Y., Cambria, E., Hussain, A., and Zhao, H. (2015). Word polarity disambiguation using bayesian model and opinion-level features. *Cognitive Computation*, 7(3).

[306] Xu, F., Uszkoreit, H., Du, Y., Fan, W., Zhao, D., and Zhu, J. (2019). Explainable ai: A brief survey on history, research areas, approaches and challenges. In *CCF international conference on natural language processing and Chinese computing*, pages 563–574. Springer.

[307] Xu, H., Peng, H., Xie, H., Cambria, E., Zhou, L., and Zheng, W. (2020). End-to-end latent-variable task-oriented dialogue system with exact log-likelihood optimization. *World Wide Web*, 23:1989–2002.

[308] Xu, P. and Hu, Q. (2018). An end-to-end approach for handling unknown slot values in dialogue state tracking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1448–1457.

[309] Xu, P. and Sarikaya, R. (2013). Convolutional neural network based triangular crf for joint intent detection and slot filling. In *2013 ieee workshop on automatic speech recognition and understanding*, pages 78–83. IEEE.

[310] Yang, X., Tang, K., Zhang, H., and Cai, J. (2019). Auto-encoding scene graphs for image captioning. In *CVF*.

[311] Ylikoski, P. (2007). The idea of contrastive explanandum. In *Rethinking explanation*. Springer.

[312] Young, T., Cambria, E., Chaturvedi, I., Zhou, H., Biswas, S., and Huang, M. (2018). Augmenting end-to-end dialogue systems with commonsense knowledge. In *AAAI*, pages 4970–4977.

[313] Young, T., Pandelea, V., Poria, S., and Cambria, E. (2020). Dialogue systems with audio context. *Neurocomputing*, 388:102–109.

[314] Young, T., Xing, F., Pandelea, V., Ni, J., and Cambria, E. (2022). Fusing task-oriented and open-domain dialogues in conversational agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11622–11629.

[315] Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

[316] Zhang, C., Li, Y., Du, N., Fan, W., and Philip, S. Y. (2019a). Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267.

[317] Zhang, D., Liu, J., Zhu, H., Liu, Y., Wang, L., Wang, P., and Xiong, H. (2019b). Job2vec: Job title benchmarking with collective multi-view representation learning. In *CIKM*.

[318] Zhang, X. and Wang, H. (2016). A joint model of intent determination and slot filling for spoken language understanding. In *IJCAI*, volume 16, pages 2993–2999.

[319] Zhang, Y., Ahmed, A., Josifovski, V., and Smola, A. (2014). Taxonomy discovery for personalized recommendation. In *Proceedings of the 7th ACM international conference on Web search and data mining*.

[320] Zhang, Y., Gan, Z., Fan, K., Chen, Z., Henao, R., Shen, D., and Carin, L. (2017). Adversarial feature matching for text generation. In *Proceedings of Conference on Machine Learning*. JMLR. org.

[321] Zhao, J., Mahdieh, M., Zhang, Y., Cao, Y., and Wu, Y. (2021a). Effective sequence-to-sequence dialogue state tracking. *arXiv preprint arXiv:2108.13990*.

[322] Zhao, W., Peng, H., Eger, S., Cambria, E., and Yang, M. (2019). Towards scalable and reliable capsule networks for challenging NLP applications. In *ACL*, pages 1549–1559.

[323] Zhao, X., Huang, W., Huang, X., Robu, V., and Flynn, D. (2021b). Baylime: Bayesian local interpretable model-agnostic explanations. In *Uncertainty in Artificial Intelligence*, pages 887–896. PMLR.

[324] Zhao, Y., Wang, Z., and Huang, Z. (2021c). Automatic curriculum learning with over-repetition penalty for dialogue policy learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14540–14548.

[325] Zhou, L., Gao, J., Li, D., and Shum, H.-Y. (2020). The design and implementation of xiaoice, an empathetic social chatbot. *Computational Linguistics*, 46(1):53–93.

[326] Zhou, Z.-H., Jiang, Y., and Chen, S.-F. (2003). Extracting symbolic rules from trained neural network ensembles. *Ai Communications*, 16(1):3–15.

[327] Žilka, L., Marek, D., Korvas, M., and Jurcicek, F. (2013). Comparison of bayesian discriminative and generative models for dialogue state tracking. In *Proceedings of the SIGDIAL 2013 Conference*, pages 452–456.