Alma Mater Studiorum - Università di Bologna

DOTTORATO DI RICERCA IN
DATA SCIENCE AND COMPUTATION

Ciclo 34

Settore Concorsuale: 01/B1 - INFORMATICA
Settore Scientifico Disciplinare: INF/01 - INFORMATICA

HOW TO EXPLAIN: FROM THEORY TO PRACTICE

Presentata da: Francesco Sovrano

Coordinatore Dottorato
Daniele Bonacorsi

Supervisore
Fabio Vitali

Co-Supervisori
Amanda Prorok
Monica Palmirani

Esame finale anno 2023

To anyone with at least a bit of madness

Abstract

Today we live in an age where the internet and artificial intelligence allow us to search for information through impressive amounts of data, opening up revolutionary new ways to make sense of reality and understand our world. However, it is still an area of improvement to exploit the full potential of large amounts of explainable information by distilling it automatically in an intuitive and user-centred explanation. For instance, different people (or artificial agents) may search for and request different types of information in a different order, so it is unlikely that a short explanation can suffice for all needs in the most generic case. Moreover, dumping a large portion of explainable information in a one-size-fits-all representation may also be sub-optimal, as the needed information may be scarce and dispersed across hundreds or thousands of pages. The aim of this work is, therefore, to investigate how to automatically generate (user-centred) explanations from heterogeneous and large collections of data, with a focus on the concept of explanation in a broad sense, as a critical artefact for intelligence, regardless of whether it is human or robotic. Our approach builds on and extends Achinstein's philosophical theory of explanations, where explaining is an illocutionary (i.e., broad but relevant and deliberate) act of usefully answering questions. Specifically, we provide the theoretical foundations of Explanatory Artificial Intelligence (YAI), formally defining a user-centred explanatory tool and the space of all possible explanations, or explanatory space, generated by it. We present empirical results in support of our theory, showcasing the implementation of new YAI tools and strategies for assessing explainability. To justify and evaluate the proposed theories and models, we considered case studies and examples at the intersection of artificial intelligence and law, particularly European legislation. Ultimately, our tools helped produce better explanations of software documentation and legal texts for humans and complex regulations for reinforcement learning agents.

Introduction

THE MAIN research question answered with this work is how to algorithmically generate user-centred and goal-oriented explanations about *something to explain* (also called *explanandum*) from a sufficiently large and heterogeneous collection of explainable information when no assumptions are available about the *recipient of the explanation* (also called *explainee*). In particular, we study what defines and constitutes *Explanatory Artificial Intelligence (YAI)* as a presentation logic capable of selecting information that can be used to explain and effectively convey knowledge to an end-user. To do so, we build on and extend philosophical theories of explanations, framing the *act of explaining* as answering questions (*plural*) with the specific intent (also called *illocution*) of concretely producing understanding in someone. We also present new theoretical models and concrete software applications to generate and assess explanations for human and artificial intelligence.

In this work, we explore various case studies and examples to justify and evaluate our proposed theories and tools. Our focus lies primarily on the intersection of artificial intelligence and law, with additional consideration given to finance, healthcare, and robotics. By applying our theory, we aim to achieve the following objectives:

- Develop explanatory software that complies with the European General Data Protection Regulation;
- Assess the compliance of software documentation in finance and health-

care against Business-to-Consumer and Business-to-Business requirements established by European legal provisions;

- Improve educational textbooks to more effectively explain legal concepts to students, with a focus on teaching them how to write a legal memorandum according to the U.S. legal system;

- Enhance the performance and adaptability of Reinforcement Learning agents, allowing for a more efficient learning process in understanding and adhering to (road) regulations.

Notably, there are two main benefits from this research. First, it can foster the deployment of automated decision-making systems in the EU landscape by defining workable methods for the production and evaluation of human-centred and lawful explanations. Second, it provides new formalisms for the design of efficient, user-centred explanatory processes, providing insights into how these can be adapted to enhance machine learning. In other words, this research is relevant to the current social, legal and technological context and addresses long-studied problems.

The production of explanations is central to the ability of human beings to make sense of reality, communicate and produce scientific discoveries. For this reason, it has been studied since ancient times[1], assuming an ever-increasing importance for various disciplines, including the sciences of education, philosophy, law and also computer science (with Artificial Intelligence).

In computer science, interest in the concept of explanation has grown with the importance of Artificial Intelligence (AI) in our society and the increasing need to explain the complexity of modern software systems. Indeed, not being able to explain the decisions provided by an automated decision-maker could have disruptive effects on the welfare of society, industry and public administration, negatively affecting the lives of billions of people. This need gave rise to Explainable Artificial Intelligence (XAI), as a discipline whose aim is to explain AI [9], and to numerous political initiatives to prevent the potential damage that a lack of explainability could have in a complex society such as ours.

Governments have started to act towards the establishment of ground rules of behaviour for complex systems. It has happened, for instance,

[1]One of the first known philosophers to (indirectly) work on the concept of explanation was Aristotle with his *theory of causation* [204].

with the enactment of the European General Data Protection Regulation (GDPR)[2], which establishes *fairness*, *lawfulness* and *transparency* as fundamental principles for data processing tools, identifying a new *right to explanation* for individuals whose legal status is affected by an algorithm. As a result, several expert groups, including those acting for the European Commission, have started asking the AI industry to adopt an ethics code of conducts as quickly as possible [43, 74], drawing a set of expectations to meet in order to guarantee citizens a right to explanation.

Today, the EU has several laws in force, which establish obligations of explainability based on who uses AI (e.g., public authorities, private companies) and the degree of automation of the decision-making process (e.g., fully or partially automated) [22]. Existing European laws (e.g., the GDPR, or the proposed Artificial Intelligence Act[3]) require AI developers to explain the logic of their software and produce ad hoc documentation.

Since, in theory, there can be an infinite number of different and heterogeneous types of explanations, identifying a minimum set of properties that all reasonable explanations should possess is not trivial. Though laws, legislators, industry and policy-makers have tried (directly or indirectly) to address this issue. For instance, the High-Level Expert Group on Artificial Intelligence (AI-HLEG) [147] was established in 2018 by the European Commission to identify a list of core ethical principles for *trustworthy AI* tools. According to the AI-HLEG, explanations should be "adapted to the expertise of the stakeholder concerned" (e.g., layperson, regulator or researcher). Moreover, they should be "highly dependent on the context" [147], putting individual's needs at the centre and indicating a preferred (user-centred) direction on how to shape explanations.

However, fully explaining complex (e.g., an AI system) or large (e.g., the Internet) amounts of information in an intuitive and user-centred manner is still an open problem. For instance, different explainees may search for and request different types of information, even if explanatory documents (e.g., books, articles, web pages, technical documentation) usually have predefined exposition and content. Different books on the same subject expound the same knowledge in different ways and with different levels

[2]Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons concerning the processing of personal data and the free movement of such data, and repealing Directive 95/46/EC, https://eur-lex.europa.eu/eli/reg/2016/679/oj

[3]https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52021PC0206

of detail. Not to mention that the background knowledge of different explainees can vary considerably, so even the most common concepts have to be explained in depth every time an explainee needs to acquire them or refresh his/her memory.

Most of the available explanatory content is static (i.e., generated once and for all, regardless of the needs and background of the readers), with information that may be scattered over hundreds or thousands of pages and indirectly dependent on external knowledge to be understood by the reader. This type of static representation in the most generic scenario is suboptimal and time-consuming because helpful information may be sparse or absent. As a metaphor, consider the goal-oriented aspects of explaining a complex project akin to searching for information about a bank robbery using the recording of a Closed-Circuit Camera (CCC) in front of the bank entrance. The fact that the CCC system can store hours of good-quality video is instrumental, but more is needed to determine the usefulness of the CCC service. For instance, investigators may know the time of the robbery but not the face of the robbers. They may know their faces but not how long they waited outside of the entrance, or the number of people that entered, or the direction they fled to, or the licence plate of the car they drove, or whether they had been there before for recognisance of the place, or even the same questions could be made not for the bank robbery but for a night burglary at the liquor store two doors down the bank. It is the investigator's specific goal, not the recording machine's quality and technicalities, that determines the questions that the CCC system must answer. Therefore, more is needed than providing 48 hours of good quality video with no tool for navigating it other than watching it at 1x speed.

In practice, the difference between *explainable information* and *explanations* becomes more apparent when the size of explainable information is sufficiently large. One can imagine many examples (besides the CCC system) where searching for an explanation is equivalent to looking for a needle in a haystack of explainable information. For example, suppose that the user of a complicated AI-based credit approval system deployed by a bank needs to know why the system rejected his/her loan application and what to do to change that outcome. In this case, the bare output of a XAI might not be enough to entirely understand how to change a loan application's outcome. At the same time, the documentation about how a credit score is computed and used for approval might be too burdensome, complicated and technical for a layperson. Indeed, the XAI might be able to

tell the applicant that she/he was rejected because of an excessive amount of "*credit inquiries*". However, it cannot tell how to reduce the number of such inquiries, nor that only the "*hard (credit) inquiries*" should be avoided, or what "*hard inquiries*" are.

In other words, static or (relatively) short explanations alone do not provide enough information to answer all the possible questions. Instead, their output needs to be reorganised and enriched with additional information. So, generally speaking, what happens is that explaining to intelligent beings (e.g., humans) is a highly challenging task, regardless of whether what has to be explained (i.e., the explanandum) is an AI, its documentation, a scientific article, a regulation, or a textbook. Additionally, the complexity of this task is increased by the elusiveness of the concept of explanation, being hard to define formally in a computer-friendly way, considering the abstract nature of current theories of explanations, which come mainly from philosophy.

In particular, the concept of explanation in philosophy started to have a more precise role in the 20th century, with the growth and development of the philosophy of science driven by Hempel and Oppenheim [88]. Eventually, this gave rise to several competing theories. Some of them focus mainly on scientific approaches and causality. For example, Causal Realism [176] frames explanations as descriptions of causes and effects, while Constructive Empiricism [210] defines explanations as descriptions of causality that are statistically plausible. Some others (e.g., Achinstein's [2]) advocate a theory of explanation that is more grounded in the way people carry out explanations [131], thus opening up to types of explanations other than causality (cf. Chapter 2).

This work starts from the work of the AI-HLEG, identifying user-centrality as the cornerstone of any framework for explanatory tools, and uses Achinstein's theory of explanations [2] from Ordinary Language Philosophy to define a model of the user-centred explanatory process. In particular, this dissertation broadly focuses on the concept of explanation as a crucial artefact for intelligence, regardless of whether it is human or robotic.

This document aims to address **five primary research questions** concerning explaining, explanations, and explainability in the context of artificial intelligence. Each research question is labelled as RQ1, RQ2, and so on, and is accompanied by a set of more specific sub-questions:

RQ1. How can one define *explaining*, *explanations* and *explainability*?

a) What are the philosophical theories of explanation that align with

the expectations set forth in laws and ethical guidelines, such as those found in European regulations like the GDPR and the proposed AI Act?

b) Can a philosophical theory be interpreted in a computer-friendly way?

RQ2. How to quantitatively evaluate *explanations* and *explainability*?

a) Is evaluating explanations the same of evaluating explainability?

b) How can a formal definition of explanation be used to construct a metric to objectively measure the degree of explainability of textual information?

RQ3. How to model an automatic (user-centred) *explanatory process*?

a) How can a philosophical theory of explanations be adapted to a computational context?

b) What are the key components of a user-centred explanatory tool, and how can the space of all possible explanations, or explanatory space, generated by such a tool be formally defined and navigated?

RQ4. How to algorithmically generate *explanations* for humans?

a) How can linguistic theories, data mining, and AI techniques be employed to implement the identified model of explanatory process?

b) How can the algorithmic explanation generation process be adapted for different domains, such as legal or educational contexts?

RQ5. Would a better understanding of what constitutes an *explanatory process* help improve artificial intelligence (i.e., machine learning)?

a) How can the identified model of explanatory process be applied to improve the learning efficiency of artificial intelligence, specifically Reinforcement Learning (RL) agents?

The **methodology** adopted to answer the above research questions comprises the following steps and publications. First, we conducted an exploratory literature review and analysis of theories of explanation in contemporary philosophy, looking for intersections with legal requirements

(stemming from the GDPR and the AI Act) and ethical guidelines (publications: [192, 196, 188, 198, 202]). We then identified a definition of explanations, explainability and explainable information, and the mechanisms for evaluating them (publications: [187, 188, 190, 191]). Next, we defined the properties and characteristics that an explanatory process should possess, defining a user-centred explanatory process and the space of all possible explanations generated by it (publications: [196, 189]). Finally, we designed a model to concretely implement the explanatory tools, thus creating different YAI for humans and artificial intelligence, evaluating the quality and generality of the model (publications: [196, 194, 195, 188, 189, 199, 190, 200, 201]).

More specifically, **the work of this thesis contributes new theory, models and concrete tools**, including:

C1. A computer-friendly extension of Achinstein's theory that starts from the definition of *explanatory illocution* as the main mechanism responsible for anticipating unformulated questions. See Chapter 3.

C2. A formal definition of explanatory tool (alternatively called YAI) and a new model, called the *SAGE-ARS*[4] *model*, to concretely implement legally compliant and user-centred YAI software by defining: *i)* the heuristics for efficiently and effectively exploring an explanatory space (ARS: Abstraction, Relevance and Simplicity); *ii)* the set of (SAGE: Sourcing, Adapting, Grounding and Expanding) commands for an explainee to interact with the explanatory process. See Chapter 5.

C3. DoX: the first metric based on Ordinary Language Philosophy to measure explainability and able to objectively quantify the Degree of Explainability of any textual information written in natural language (e.g., English). See Chapter 4 and Chapter 8.

C4. YAI4Hu: an implementation of the SAGE-ARS model (evaluated with two user studies involving more than 190 participants) based on AI for question-answering, for better explaining AI systems and software documentation (in accordance with the GDPR). See Chapter 6 and Chapter 7.

C5. Two new algorithms for question-answering (SyntagmTuner and DiscoLQA), on technical documents, without expensive datasets and train-

[4] "Ars" means art in Latin.

ing procedures, together with a new evaluation dataset for answer retrieval that includes more than 70 questions and 200 answers on 6 different European norms. These algorithms are based on linguistic theories of discourse [157] and sentential meaning representation [15]. See Chapter 9.

C6. YAI4Edu: an intelligent interface that extends YAI4Hu and uses DoX, SyntagmTuner and DiscoLQA to improve the explanatory power of a textbook for teaching how to write a legal memorandum according to the U.S. legal system. See Chapter 10.

C7. XAER: a novel AI mechanism using the ARS heuristics for intelligently explaining complex regulations and reward functions more efficiently and effectively to single-agent automated decision makers based on seminal off-policy RL algorithms (i.e., DQN, TD3, SAC). See Chapter 12.

C8. DEER: an extension of XAER for multi-agent RL algorithms. See Chapter 13.

All these contributions demonstrate that user-centred explanations are better understood as individual, goal-driven paths within a vast, possibly unlimited explanatory space. Moreover, each path's direction, length and components depend directly and substantially on the type of need, objective and background of the explainee for whom it is intended.

Considering the nature of the contributions, **this dissertation is structured in three main parts**: *i)* theoretical contributions; *ii)* explanatory tools for humans; *iii)* explanatory tools for robots.

In **Part I (Theoretical Contributions)** we introduce the theoretical underpinnings of YAI, providing an answer to *RQ1*, *RQ2* and *RQ3*. To do so, in Chapter 1 we first perform an in-depth overview of the right to explanation of the GDPR, the AI-HLEG guidelines for trustworthy AI, and the role of explainability in the proposed AI Act. Thus we identify some types of explanations required by the law and the property of user-centrality as one of the main characteristics of a good explanatory process. In Chapter 2 we provide the necessary background to understand which philosophical definitions of explanations are compatible with the requirements set by European regulations. We focus on Achinstein's theory and explain the difference between *explainable information* and *explanation*. Soon after, in Chapter 3, we discuss how Achinstein's theory should be adapted to use

for the design of practical explanatory software. To do so, we introduce the concepts of *explanatory illocution* and *archetypal questions*, also explaining why *usability metrics* are a good choice for evaluating explanations and illocution in this case. Next, in Chapter 4 we show how the identified explanation theory can be used to quantify the Degree of Explainability (DoX) of information objectively. We define DoX as an explainability metric by formalising the concept of explanatory illocution. In that chapter, we also discuss how DoX can help to verify law compliance better than other explainability metrics. Finally, Chapter 5 defines YAI as a discipline separate from XAI, discussing the differences between user-centred and one-size-fits-all explanations. Here we introduce the concept of *explanatory space* as a *hypergraph*[5] of explanations, modelling an explanatory tool as a mechanism for decomposing such space into a sequence of information (i.e., a tree-like structure). We also propose the SAGE-ARS model for generating user-centred sequence decompositions, showing a proof of concept of how it can be used to create YAI software compliant with the GDPR.

In **Part II (Explanatory Tools for Humans)** we answer *RQ4*, evaluating the theoretical models discussed in Part I through concrete software applications for humans. In particular, in Chapter 6, we explain how to use AI algorithms for answer retrieval and information extraction to implement YAI4Hu, an example of YAI system based on the SAGE-ARS model. In Chapter 7, we experimentally validate YAI4Hu, showing that it is statistically more effective at explaining than one-size-fits-all explanatory tools. We prove it by performing two between-subjects user studies involving hundreds of participants and comparing the user-centrality (measured in terms of *usability*) of the explanatory systems. As case studies for the evaluation, we considered two scenarios where the documentation and the automated decisions of AI systems for credit approval and heart disease prediction are explained to comply with the law or to help system users achieve their goals. Then, Chapter 8 describes how to use the technology behind YAI4Hu to implement DoXpy, a software implementation for estimating DoX scores. DoXpy is used to demonstrate through experiments that explainability changes according to DoX and that DoX is a reasonable metric for explainability. In Chapter 9, we deal with how to specialise existing question-answering algorithms to work on particular collections of texts written in technical languages such as legal English, without resort-

[5] A hypergraph is a generalization of a graph in which an edge (or hyperedge) can join any number of vertices [30].

ing to expensive training datasets and procedures. Specifically, Chapter 9 presents Q4EU (a dataset for evaluating answer retrieval on European legislation), SyntagmTuner (a strategy for combining shallow and deep learning approaches to cope with the lack of a training set) and DiscoLQA (a mechanism that uses discourse theory to help an answer retriever identify and isolate the unique discursive constructs used by a technical language). Finally, in Chapter 10, we present YAI4Edu, an intelligent interface that uses DoX, SyntagmTuner and DiscoLQA to build intelligent textbooks for education automatically. We show how YAI4Edu can increase the explanatory power of a textbook, anticipating the needs of an explainee, by automatically identifying a set of helpful (implicit) questions that a reader might have about the textbook and reorganising the contents of the textbook accordingly. We also evaluate YAI4Edu with a case study in legal writing and a within-subjects user study involving more than one hundred English-speaking students.

In **Part III (Explanatory Strategies for RL Agents)** we answer *RQ5*, showing how to implement the SAGE-ARS model on seminal RL algorithms. In Chapter 11, we provide the technological background to understand what RL is and what it means to explain to an RL agent. Then, in Chapter 12, we present XAER, an YAI mechanism for explaining to (off-policy) RL agents through a training procedure called experience replay. We show how XAER can model the experience buffer of an RL agent as an explanatory space, extracting explanatory sequences of experience through the ARS heuristics so that the agent can learn faster and (sometimes) even better. As a case study, we also show how XAER can explain dense road rules at different levels of complexity. Finally, in Chapter 13, we discuss the main problems associated with using a technique such as XAER in multi-agent RL, thus proposing DEER as a solution capable of scaling XAER to a larger number of agents.

Eventually, we conclude the dissertation with a brief analysis of the contributions, discussing the answers to each research question. We summarise the overall results obtained, defending the generality of the theoretical and algorithmic models and arguing that they are generic enough to broadly capture the nature of explanations as we have tested them not only with humans but also with artificial intelligence.

Contents

Part I

Models and Theoretical Contributions

Summary

In this part of the thesis, we provide the theoretical background and contributions to answer the first three research questions set out in the introduction: *i)* what is meant by explaining, explanation and explainability, *ii)* how to quantitatively evaluate explanations and explainability, *iii)* and how to model an algorithmic explanatory process.

As already anticipated, we begin from Chapter 1 with an analysis of existing legal and ethical interpretations of explanations, focusing on European regulations such as the GDPR and the proposed AI Act. Furthermore, we study the work of the High-Level Expert Group on Artificial Intelligence, an expert group of the European Commission that has proposed ethical guidelines for trustworthy AI. Accordingly, we identify user-centred explanatory tools as essential for reliable AI. In Chapter 2, we look for contemporary philosophical theories of explanations that can be compatible with the expectations outlined in the laws and ethical guidelines mentioned above. We do this on the premise that the idea of a user-centred explanatory process has its roots in (contemporary) philosophy.

Among these philosophical theories, we identify Achinstein's, from Ordinary Language Philosophy, as a suitable candidate for defining an explanation as an illocutionary (i.e., broad, but relevant and deliberate) act of answering questions in a useful way for the explainee. Furthermore, we

The content of Part I is a reworking and extension of the following articles by the same author of this thesis: [192, 196, 188, 187, 198, 202, 190, 191, 189].

differentiate explanations from explainable information, arguing that there is a subtle but critical difference between them. Since our primary goal is to model explanatory software for the automatic generation of user-centred explanations, in Chapter 3, we extend Achinstein's theory by defining explanatory illocution in a computer-friendly way. Then, in Chapter 4, we show how a formal definition of explanatory illocution can help construct a metric to objectively measure the degree of explainability of textual information. We also discuss how such a metric could facilitate the assessment of compliance with the proposed European AI Act. Finally, in Chapter 5, we provide the theoretical basis of Explanatory Artificial Intelligence, formally defining a user-centred explanatory tool and the space of all possible explanations, or explanatory space, generated by it.

CHAPTER 1

Legal Background: AI, Explanations, Ethics and Law

This chapter provides some background information to understand the existing legal requirements and ethical guidelines on explanation in Europe. We start with the famous *right to explanation* introduced by the GDPR and then move on to the work of the AI-HLEG and the forthcoming Artificial Intelligence Act. The main objective is thus to understand what characteristics explanations should have to comply with ethics and (European) law.

1.1 The Right to Explanation

The GDPR is a relevant 2016 European regulation on protecting personal data and related rights and freedoms. Since the GDPR is technology-neutral, it does not directly refer to AI. However, several provisions are highly relevant to the use of AI (or any other software) for automated decision-making processes. For instance, according to the Information Commissioner's Office of the United Kingdom [97], the most important

of these provisions are:

- Article 5(1) point (a), that requires personal data processing to be fair, lawful, transparent, necessary and proportional.
- Article 12, which defines the obligations for transparent communication and the modalities for data subjects to exercise their rights.
- Articles 13, 14 and 15, that give individuals the right to be informed of solely automated decision-making, meaningful information about the logic involved, and the significance and envisaged consequences for the individual.
- Article 22, that gives individuals the right not to be subject to a solely automated decision producing legal or similarly significant effects.
- Article 22(3), that obliges organizations to adopt suitable measures to safeguard individuals when using solely automated decisions, including the right to obtain human intervention, to express his or her view, and to contest the decision.

Altogether, these provisions are the source of a debate over the so-called right to explanation [105]. More specifically, the *right to explanation* is a right that individuals might exercise when their legal status is affected by a solely automated decision-making process.

The reasons for decisions must be adequately explained to put European citizens in a position to challenge an automated decision and thus exercise their right to contest it. Specifically, in case of contract or consent, Article 22, paragraph 3 introduces the "right to obtain human intervention on the part of the controller, to express his or her point of view and to contest the decision". Here explanations seem to be provided only after decisions have been made (*ex-post* explanations) and are not a required precondition to protest decisions. Instead, Articles 13, 14 and 15 of the GDPR require an overview of a system prior to processing (*ex-ante* explanations), with the obligation to inform about the "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 (Recital[1] 63), as well as the significance and the envisaged consequences of such processing for the data subject". In other words, the GDPR defines (indirectly)

[1] A Recital is supposed to cast light on the interpretation to be given to a legal article/rule but it cannot, per se, constitute such a rule [105].

two modalities of explanation: explanations can be offered before (*ex-ante*; Articles 13, 14 and 15) or after decisions have been made (*ex-post*; Article 22, paragraph 3).

It is not clear from the GDPR whether a "right to an explanation" should imply user-centred personalised explanations. Regardless of the answer, according to the GDPR, the data controller must provide "meaningful information about the logic involved" in an automated decision, explaining it. See Art. 13(2)(f), 14(2)(g), 15(1)(h).

For each modality, the GDPR defines goals and purposes of explanations, thus providing a set of explanatory contents. Additionally, the white paper on Artificial Intelligence [53] by the European Commission stressed the need to monitor and audit not only automated decision-making algorithms but also the data records used for training, developing, and running, the AI systems in order to fight the opacity and to improve transparency. Hence, from a technical point of view, technology-specific information must be considered to meet the GDPR explanation requirements fully.

Fundamentally, *ex-ante*, we should provide information that guarantees the transparency principle, describing: *i)* the algorithms and models pipeline composing the automated decision-making process; *ii)* the data used for training (if any), developing and testing the automated decision-making process; *iii)* the background information (e.g., the jurisdiction of the automated decision-making process); *iv)* the possible consequences of the automated decision-making process on the specific data subject.

Ex-post the data subject should be able to contest a decision fruitfully, so he/she should be given access to: *i)* the justification about the final decision; *ii)* the run-time logic flow (causal chain) of the process determining the decision; *iii)* the data used for inferring; *iv)* information (metadata) about the physical and virtual context in which the automated process happened.

Therefore, the GDPR draws a set of expectations to meet in order to guarantee the right to explanation. These expectations are meant to define the goal of explanations and, thus, explanatory content that may evolve together with technology. This explanatory content identifies at least three different types of explanations: causal, descriptive, justificatory. These are the minimal explanations required for explaining automated decision-making systems **under the GDPR**. In fact, in the case of GDPR, we have that:

- **Descriptive explanations** are primarily required in the *ex-ante* phase

to explain business models, the possible effects of automated decision-making systems on a user, and the characteristics and limitations of the algorithms.

- **Causal explanations** are primarily required in the *ex-post* phase to explain the causes of a solely automated decision.
- **Justificatory explanations** are required in both the *ex-ante* and *ex-post* phases to justify decisions, for example, through permissions and obligations.

The explanations mentioned above can be provided to the user through one or more explanatory tools as part of the whole AI system. Nonetheless, despite the variety of required explanatory contents, the GDPR does not specify what qualifies and formally constitutes an explanation. In the likely attempt to overcome this issue, the Think Tank of the European Parliament [61] listed the following qualities that a reasonable explanation should possess: intelligibility, understandability, fidelity, accuracy, precision, level of detail, completeness and consistency. Nevertheless, the debate is still open as to whether explanations should be personalised and user-centred. Indeed, Article 22 lends itself to different interpretations [149, 151] as to whether providing personalised explanations is mandatory or just good practice. To this end, Recital 71 provides interpretative guidance of Article 22. However, two items are missing in Article 22 relative to Recital 71: the provision of "specific information" and the "right to obtain an explanation of the decision reached after such assessment". The second omission, in particular, raises the issue of whether controllers are required by law to provide an individualised explanation. This issue is partially tackled by the guidelines of the High-Level Expert Group on Artificial Intelligence (endorsed by the European Commission), giving further reason to believe that there is the intention to prefer user-centred explanations as soon as the technology is mature enough to guarantee them. Instead, Recital 63 requires *ex-ante* that the data subject should have the right to know and obtain communication, in particular about "the logic involved in any automatic personal data processing".

1.2 Transparency and Ethical Guidelines for Trustworthy AI

The High-Level Expert Group on Artificial Intelligence (AI-HLEG), independent of political parties and representing academia and industry, was commissioned by the European Union to identify a set of "Ethical Guidelines for Trustworthy AI". These guidelines were published in April 2019 [147] and then finalised in 2020 by the same AI-HLEG into a self-assessment checklist called Assessment List for Trustworthy AI (ALTAI). Notably, the guidelines identified by the AI-HLEG are not enforced by law like the GDPR. However, they are more specific than the GDPR about the properties that a good explanation and a good explanatory tool should have for reliable AI.

According to the AI-HLEG, good explanations should be "adapted to the expertise of the stakeholder concerned" (e.g., layperson, regulator or researcher) and "highly dependent on the context", putting individual's needs at the centre and indicating a preferred (user-centred) direction on how to shape explanations. The AI-HLEG vision of a user-centred AI seems to incorporate the GDPR principles. It tries to expand them into a broader framework based on four consolidated ethical principles (i.e., respect for human autonomy, prevention of harm, fairness and explicability) from which seven key requirements for trustworthy AI are derived, including: human agency and oversight, transparency (including traceability, explainability and communication), diversity, non-discrimination and fairness (including the avoidance of unfair bias, accessibility and universal design, and stakeholder participation), accountability (including auditability, minimisation and reporting of negative impact, trade-offs and redress).

The ethical principle of *explicability* [74] is typically associated with the requirements of *transparency* and *accountability*, taking clear inspiration from Articles 13, 14, 15 and 22 of the GDPR. Hence, in a way, the AI-HLEG applies to AI the technologically neutral GDPR by defining relevant guidelines on how transparency can be achieved in trustworthy systems, also through accessibility and universal design. The *transparency* requirement, in particular, covers the transparency of relevant elements for an AI system (data, algorithms and business models), including:

- *Traceability*: "the datasets and the processes that yield the AI system's decision, including data gathering and labelling and the algorithms used, should be documented".

- *Explainability*: is defined as "the ability to explain both the technical processes of an AI system and the related human decisions (e.g., application areas of a system)", thus explicitly implying also "business model transparency".

- *Communication*: "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".

1.3 The Role of Explainability in the Proposed AI Act

The discussion towards "explainability and law" has departed from the contested existence of a right to explanation in the GDPR to embrace contract, tort, banking law [85], and judicial proceedings [69]. While these legal sectors present significant differences in the applicable law and jurisdiction, they all highlight the significance of algorithmic transparency within existing legal sectors.

On April 21, 2021, the European Commission proposed the AI Act, the first legal framework on AI to address the risks posed by this emerging method of computation. The proposed AI Act considers not only machine learning but expert systems and statistical models long in place. Differently from other domains, the AI Act is specific to AI systems and requires an ad hoc discussion rather than the framing of these systems in the discussion of other legal domains. It is because AI technologies are not placed within an existing legal framework (e.g., banking), but the whole legal framework (i.e., the AI Act) is built around AI technologies. Under the proposed AI Act, new obligations are set to ensure transparency, lawfulness and fairness. Their goal is to establish mechanisms to ensure quality at launch and throughout the whole life cycle of AI-based systems, thus ensuring legal certainty that encourages innovation and investments on AI systems while preserving fundamental rights and values. Specifically, the AI Act

The work presented in Section 1.3 and Section 4.2 was developed in collaboration with Salvatore Sapienza from the University of Bologna [198]. *S. Sapienza*: legal analysis constituting this Section 1.3, part of the introduction of [198]. *S. Sapienza* and *F. Sovrano*: definition of the four main principles for explainability metrics introduced in Section 4.2. *F. Sovrano*: the remaining part of [198], including the analysis of philosophical theories of explanation, all the tables and the analysis of explainability metrics.

sets some "new" minimum requirements of *explicability* (transparency and explainability) for a list of AI systems labelled as *high-risk* in Annex IV. These requirements include many technical explanations of different types.

If explainability is often instrumental to achieving some legislative goals, it could likely be meant to foster specific regulatory purposes also under the AI Act. From the joint reading of a series of provisions, it will be argued that explainability in the AI Act is both *user-empowering* and *compliance-oriented*: on the one hand, it serves to enable users of the AI system to use it correctly; on the other hand, it helps to verify adequacy to the many obligations set by the AI Act. Indeed, Recital 47 and Art. 13(1) state that high-risk AI systems shall be designed and developed so that their operation is comprehensible by the users. They should be able to interpret the system's output and use it appropriately. This is a form of *user-empowering* explainability. Then, the second part of Art. 13 specifies that "an appropriate type and degree of transparency shall be ensured, with a view to *achieving compliance* (emphasis added) with the relevant obligations of the user and of the provider [...]". This provision specifies that these explainability obligations (i.e., transparent design and development of high-risk AI systems) are *compliance-oriented*. The twofold goal of Art. 13(1) is then echoed by other provisions. As regards the user-empowering interpretation, Art. 14(4)(c) relates explainability to "human oversight" design obligations. These measures enable the individual supervising the AI system to interpret its output correctly. Moreover, this interpretation shall put him or her in the position to decide whether it might be the case to "disregard, override or reverse the output", Art. 14(4)(d).

The compliance-oriented explainability interpretation becomes evident in the technical documentation to be provided according to Article 11. Compliance is based on a presumption of safety if the system is designed according to technical standards (Art. 40), conformity with which is documented. In contrast, third-party assessment only appears after placing on the market or in specific sectors (see Chapter IV). The contents of the dossier are those detailed in Annex IV. Among other things, Annex IV(2)(b) include "the design specifications of the system, namely the general logic of the AI system and the algorithms" among the information to be provided to show compliance with the AI Act before placing the AI system in the market. Hence, the system should be explainable in a manner that allows an evaluation of conformity by the provider in the first instance and, when necessary, by post-market monitoring authorities. Since the general approach

taken by the proposed AI Act is a risk-reduction mechanism (Recital 5), this form of explainability is ultimately meant to minimise the level of potential harmfulness of the system.

User-empowering and compliance-oriented explainability overlap in Art. 29(4). When a risk is likely to arise, the user shall suspend the use of the system and inform the provider or the distributor. This provision entails understanding the system's working (in real-time) and making predictions on its output. Suspending in the case of likely risk is the overlapping between the two nuances of explainability: the user is empowered to stop the AI system to avoid contradicting the rationale behind the AI Act, i.e., risk-minimisation. Unlike the GDPR, no explicit provision enables the person affected by the system to exercise rights against the provider or the system user or to access explanations about how the system works. Moreover, explainability obligations are solely limited to high-risk AI systems: medium-risk (Art. 52) follows "transparency obligations" that consists of disclosing the artificial nature of the system in the case of chat-bots, the exposition to specific recognition systems, the "fake" nature of the image, audio or video content.

CHAPTER 2

Philosophical Background: Explanations in Contemporary Philosophy

As discussed in Chapter 1, the AI-HLEG tries to extend the GDPR expectations, targeting AI and giving further guidelines: accessibility and universal design should be a requirement for trustworthy AI, with user-centrality at the core. Furthermore, the proposed AI Act suggests that explainability (for high-risk systems) should be user-empowering, enabling the user to suspend the use of the system whenever it is no more compliant with the law. Consequently, one of the goals of the analysed legal and ethical frameworks is to put users in control and at the centre of explanatory processes. This implies that law-compliant explanations are more than just about shedding light on the chain of causes and effects of particular events. They should also justify and describe (e.g., data, processes, decisions) in a meaningful way (cf. Section 1.1). Importantly, this idea of a user-centred explanatory process not limited to causality is familiar to and finds its roots in contemporary philosophy, e.g., in the theories coming from Ordinary Language

Philosophy (i.e., Achinstein's [2]) and Cognitive Science (i.e., Holland's [91]).

Therefore, in the following sections, we will briefly summarise several recent and less recent approaches to the theories of explanation and explainability to highlight major philosophical understandings. We will mainly focus on Carnap's and Achinstein's.

2.1 Philosophical Definitions of Explainability and Explanation

If *explainability* is "the potential of information to be used for explaining", we envisage that a proper understanding of how to measure explainability must pass through a thorough definition of what constitutes an explanation and of the act of explaining.

In 1948 Hempel and Oppenheim published their "Studies in the Logic of Explanation" [88], giving rise to what is considered the first theory of explanation: the deductive-nomological model. After this work, many modified, extended, or replaced this model, which was considered fatally flawed [33, 176]. Indeed, Hempel's epistemic theory of explanations is not empiricist: it is concerned (mistakenly) only with logical form, so an explanation can be such regardless of the actual processes and entities conceptually required to understand it. Several more modern and competing theories of explanation have been the result of this criticism [131]. For example, Salmon's realist theory [176], called Causal Realism, emphasises that actual processes and entities are conceptually necessary to understand precisely why an explanation works. Instead, the Constructive Empiricism of Van Fraassen [210] relies more on a Bayesian interpretation of probability, framing explanation as a creative process of building models that are likely true.

In contrast to these theoretical and primarily scientific approaches, other philosophers have favoured a theory of explanation that is more grounded in how people perform explanations [131]. For example, Achinstein's theory [2], based on Ordinary Language Philosophy, emphasises the communicative or linguistic aspect of an explanation and its usefulness in answering questions and fostering understanding between individuals. The theory of Holland et al. [91] instead, based on Cognitive Science, frames the process of explaining as a purely cognitive activity and explanations as a certain kind of mental representation. Conversely, Sellars [183] suggests a differ-

ent way of thinking about the epistemic meaning of the explanatory act, making it more of a utilitarian process of constructing a coherent belief system.

In particular, Hempel's, Salmon's, and Van Fraassen's theories frame the act of explaining more as a *locutionary act* [12], whereby an explanation is such because it utters something. Differently, Achinstein's theory explicitly frames explaining as an *illocutionary act* [12] so that an explanation is such because of the intention to explain. The theories of Holland and Sellars, on the other hand, frame explaining more as a *perlocutionary act* [12], thus with an explanation being such because of the effects it produces in the interlocutor.

Thus, each of these theories devises different definitions of explanation and explainability, sometimes in a complementary way. A summary of these definitions is shown in Table 2.1, shedding light on the fact that there is no complete agreement on the nature of explanations. Nevertheless, according to [131], fundamental disagreements on the nature of explanations are just of two types, metaphysical and meta-philosophical, and mainly unrelated to their *logical* and *cognitive* structure. This gives room to understandings of "explanations" that may be complementary, some focusing more on cognition and others on logic.

If we analyse the specific features of these philosophical theories, we can discover that many of them are explainee-centred (or user-centred). This means that they involve customising explanations for specific explainees. Interestingly, most of them also envisage the process of answering questions as part, or foundation, of the act of explaining. While Causal Realism and Constructive Empiricism are rooted in causality, Ordinary Language Philosophy, Cognitive Science and Scientific Realism study explaining as a possibly iterative process involving broader forms of question-answering. In particular, Cognitive Science and Scientific Realism focus more on the effects of an explanation on the explainees rather than on the structure of the explanation itself.

We observe that when explaining is not considered a locutionary act and thus it is intended to meet someone's needs, explainability differs from explaining. As a result, many philosophical traditions offer definitions of "explainable information" that slightly differ from those of "explanation", as shown in Table 2.1. Indeed, pragmatically satisfying someone (e.g., user-centrality) is achieved when explanations are designed for a specific person or audience. This implies that the same explainable pieces of information

Table 2.1: ***Philosophical definitions of explanation and explainable information.*** *In this table, we summarise the definitions of explanation and explainable information for each one of the identified theories of explanations.*

Theory	Explanations	Explainable Information
Causal Realism [176]	Descriptions of causality, expressed as chains of causes and effects.	What can fully describe causality.
Constructive Empiricism [210]	Contrastive information that answers `why` questions, allowing one to calculate the probability of a particular event relative to a set of (possibly subjective) background assumptions.	What provides plausible answers to contrastive `why` questions.
Ordinary Language Philosophy [2]	Answers to questions (not just `why` ones) given with the explicit intent of producing understanding in someone, i.e., the result of an illocutionary act.	What can be used to pertinently answer questions about relevant aspects with *illocutionary force*.
Cognitive Science [91]	Mental representations resulting from a cognitive activity. They are information which fixes failures in someone's mental model.	What can have a *perlocutionary effect*, fixing failures in someone's mental model.
Naturalism and Scientific Realism [183]	Information which increases the coherence of someone's belief system, resulting from an iterative process of confirmation of truths aimed at improving understanding.	What can have a *perlocutionary effect*, increasing coherence of someone's belief system.

can be presented and re-elaborated differently across different individuals as different explanations. The type and order of explainable information matter and directly impact the quality of the resulting explanations. In simpler terms, not every combination of explainable information qualifies as an explanation according to illocutionary and perlocutionary theories.

These theories are rooted in the intent of the explainer and the subsequent effect on the listener. For instance, illocutionary theories such as Achinstein's emphasize the explainer's objective of promoting understanding. Conversely, perlocutionary theories, like Holland's, concentrate on

the actual influence explanations have on the listener. Thus, while illocutionary explanations aspire to generate a perlocutionary effect, the desired outcome is not always guaranteed. Achinstein's theory, while supporting a user-centred approach, doesn't underscore the role of the user as markedly as Holland's theory or other theories that perceive explanations as perlocutionary acts. From an illocutionary standpoint, the true intent to clarify can render information as an explanation. However, perlocutionary theories argue that an explanation is one that accomplishes its intended effect on the recipient. This divergence between intention and result emphasizes the intricate nature of explanations, both theoretically and practically.

From a pragmatic viewpoint, it is crucial to recognize that the explainer cannot fully anticipate or control the recipient's interpretation of the provided explanation or entirely comprehend their past experiences. Consequently, the intended outcomes may not always align with the actual results. This discrepancy makes illocutionary theories of explanations seem more practical, and thus more suitable for implementation in software applications. However, this is not to suggest that under Achinstein's theory an explanatory act cannot be perlocutionary. Rather, it posits that the fundamental action required to classify information as an explanation differs from other theories.

2.2 Explainability According to Ordinary Language Philosophy

In 1983, Achinstein was one of the first scholars to analyse the process of generating explanations, introducing his philosophical model of a pragmatic explanatory process. According to the model, explaining is an illocutionary act coming from a clear intention of producing new understandings in an explainee by providing a correct content-giving answer to an open-ended question. In particular, *explanatory illocution* is the deliberate intent of producing understandings [12]. Therefore, according to this view, answering by "filling the blank" of a pre-defined template answer prevents the act of answering from being explanatory by lacking illocution. These conclusions are straightforward and explicit in Achinstein's last works [3], consolidated after a few decades of public debates.

More precisely, according to Achinstein's theory, an explanation can be summarized as a pragmatically correct content-giving answer to questions of various kinds, not necessarily linked to causality. In some contexts,

pointing out logical relationships may be the key to making the person understand. In other contexts, pointing out causal connections may be sufficient. In still other contexts, other types of information may be needed. This is justified by the critical assertion that explanations have a pragmatic character so that what exactly has to be done to make something understandable to someone may (in the most generic case) depend on the interests and background knowledge of the person seeking understanding [66].

In this sense, the strong connection of Achinstein's theory to natural language and (natural) users is quite evident, for example, in the **Achinsteinian concepts** of:

- **Ellipses** or **elliptical information** [3, pp. 112-114]: intended as an explanation that is purposely shrunk to a very minimal sentence to avoid information that might be redundant for the explainee (i.e., for his/her background knowledge or common-sense).

- **U-restrictions** [3, pp. 114-119]: the meaning of an utterance/explanation u is restricted to the common interpretation of it, usually defined by grammar or rhetoric.

Indeed, according to Achinstein [2, pp. 48-53] "S explains Q to E by uttering U" is true if and only if either:

- U is constructed in a way that allows anyone to easily *restrict* (i.e., disambiguate, interpret) the meaning of U to that of a sentence expressing a complete content-giving proposition for Q.

- U is elliptical (or an *ellipsis*): it is enough for the specific E to understand a sentence expressing a complete content-giving proposition for Q.

In other words, Achinstein's definition of explanations considers not only the typical omissions of content possible under grammar and rhetoric (e.g., co-references, anaphora). It also considers all those omissions of information used to simplify an explanation, reducing the amount of redundant information for a specific explainee or common sense.

Despite this deep connection to natural (non-formal) language, Achinstein does not reject at all the utility of formalisms, hence suggesting the importance of following *instructions* (protocols, rules, algorithms) for correctly explaining some specific things within specific contexts. In this sense, Achinstein's concept of instructions [2, pp. 53-56] could be usefully

adopted to address the question of how deep or broad explanations should go. *Instructions* are "rules imposing conditions on answers to a question", or also a mechanism to check whether an answer is correct[1] in a given context. For example, they might be framed with legal requirements (as in the case of XAI [22]), ethical guidelines (i.e., [147]), or mathematics.

2.3 Adequacy of Explainability: Carnap's Criteria

In philosophy, the most important work about the criteria of adequacy of explainable information is likely to be Carnap's [42]. Even though Carnap studies the concept of *explication* rather than that of explainable information, we assert that they share a common ground making his criteria fitting in both cases. *Explication* in Carnap's sense is the replacement of a somewhat unclear and inexact concept, the *explicandum*, by a new, clearer, and more exact concept, the *explicatum*[2], and this is exactly what information does when made explainable.

Carnap's main criteria of *explication adequacy* [42] are: *similarity*, *exactness* and *fruitfulness*[3]. *Similarity* means that the explicatum should be *detailed* about the explicandum, in the sense that at least many of the intended uses of the explicandum, brought out in the clarification step, are preserved in the explicatum. In contrast, *exactness* means that the explication should be embedded in some sufficiently *clear* and exact linguistic framework. Instead, *fruitfulness* implies that the explicatum should be *useful* and usable in a variety of other *good* explanations (the more, the better).

Carnap's adequacy criteria possess preliminary characteristics for any information to be adequately considered explainable. Interestingly, the property of *truthfulness* (being different from exactness) is not explicitly mentioned in Carnap's desiderata. That is to say that explainability and truthfulness are complementary but different, as also discussed by Hilton [89]. An explanation is such regardless of its truth (high-quality but ultimately false explanations exist, especially in science). Vice versa, highly correct information can be inferior at explaining.

[1] Please note that Achinstein stresses that a correct answer does not necessarily produce understanding. So, correctness is not sufficient for an answer to be an explanation.

[2] *Explicatum* means "what has been explained", in Latin.

[3] Carnap also discussed another desideratum, *simplicity*. However, this criterion is presented as subordinate to the others (especially exactness).

CHAPTER 3

A Computer-friendly Interpretation of Illocution in Explanations

Among the philosophical theories not limited to causality (i.e., Ordinary Language Philosophy, Cognitive Science and Scientific Realism; cf. Section 2.1), the only one that is open to non-subjective (thus reproducible) evaluations of explainability is Ordinary Language Philosophy. Indeed, Cognitive Science and Scientific Realism frame explaining more as a *perlocutionary act* than an illocutionary one, imposing a direct measurement of the effects that explainable information has on people (i.e., on average). This suggests that Achinstein's theory might be the most suitable candidate to provide the theoretical background needed to objectively assess explainability (e.g., for law compliance) and algorithmically identify the best explainable information that an explanatory process should use for an explanation. Furthermore, Achinstein's theory is founded on a question-answering process, and computer scientists and engineers have already discovered how to automate such a process. Differently from other theories, this would make Achinstein's theory concretely implementable in real software applications as soon as the role of illocution (i.e., a deliberate intent

of producing the "conventional consequences" of the act [12]) can be better formalised.

Therefore, in this chapter, we will elaborate on how explaining as answering questions is widespread in computer science, motivating why Achinstein's theory is not far from the current state of the art. Next, we will discuss how to interpret the concept of *explanatory illocution* in a computer-friendly way. To do so, we will also provide the notion of *archetypal question*, explaining why *usability metrics* are a good choice for evaluating explanations and explanatory illocution.

3.1 Explaining as Answering Questions in XAI and Computer Science

Computer science, through XAI, has long studied the topic of explanations and how to generate them (e.g., for explaining complex software computations, for law compliance), frequently drawing from philosophy and social sciences [138]. Two distinct types of explainability are predominant in the literature of XAI: rule-based and case-based. Rule-based explainability is when explainable information is a set of formal logical rules describing information related to cause and effects. For example, the inner logic of a model, its causal chain, how it behaves, why that output gave the input, and what would happen if the input were different. While case-based explainability is when the explainable information is a set of input-output examples (or counter-examples) meant to give an intuition of the model's behaviour. For example, counterfactuals, contrastive explanations, or prototypes[1].

The idea of answering questions as explaining is familiar to XAI and compatible with everyone's intuition of what constitutes an explanation. In fact, despite the different types of explainability one can choose, it is always possible to frame the information provided by explainability with one or (sometimes) more questions. In particular, it is common to many works in the field [167, 125, 138, 80, 62, 217, 164, 99, 127] the use of generic (e.g., `why`, `who`, `how`, `when`) or more punctual questions to clearly define and describe the characteristics of explainability [124]. For example, Lundberg et al. [126] assert that the local explanations produced by their TreeSHAP (a XAI algorithm for estimating the importance of features as

[1]Prototypes are instances of the ground-truth considered similar to a specific input-output for which the similarity explains the model's behaviour.

input to an AI model) might enable "agents to predict `why` the customer they are calling is likely to leave" or "help human experts understand `why` the model made a specific recommendation for high-risk decisions". Similarly, Dhurandhar et al. [62] state that they designed CEM (a XAI algorithm for the generation of counterfactuals and other contrastive explanations) to answer the question "`why` is input x classified in class y?".

These are just some examples, among many, of how Achinstein's theory of explanations is already implicit in existing XAI literature, highlighting how deep the connection between answering questions and explaining is in this field. A connection that has been implicitly identified also by Lim et al. [125], Miller [138] and Gilpin et al. [80] that analysing XAI literature were able to hypothesise that a good explanation, about an automated decision-maker, answers at least the following questions:

- What did the system do?
- Why did the system do P?
- Why did the system not do X?
- What would the system do if Y happens? ,
- How can one get the system to do Z, given the current context?
- What information does the system contain?

In particular, from a preliminary analysis, it appears that most classical XAI algorithms focus more on the production of explainable software and explanations that generally follow a one-size-fits-all approach, answering one (or sometimes a few) predefined questions well. However, one-size-fits-all explanations tend to lack user-centrality, usually failing to answer all the questions an explainee might have. This is also suggested by Liao and Varshney [123], who show that no single XAI seems to be able to cover all identified user needs and that various XAI algorithms may be needed to explain a system better. Indeed, users' needs in terms of explainability are multiple and challenging to capture [124], e.g., they may concern terminology, system performance, system outputs, and inputs.

User-orientedness is challenging and sometimes not connected to the primary goal of XAI: "opening the black box", i.e., understanding how and why an opaque AI model works. Compared to creating explanations for AI experts, generating user-centred explanations is more challenging since, in

many cases, it is unrealistic to ask them to interpret internal parameters, and complex computations of AI models [100]. For example, a layperson trying to receive a loan might be interested in knowing that her/his application was rejected (by an AI) mainly because of a high number of inquiries on her/his accounts (as TreeSHAP or CEM can tell). However, this information alone may not be enough for her/him to reach her/his goals. These goals may be out of the scope of XAI, as to understand: how to effectively reduce the number of inquiries in order to get the loan, which types of inquiries may affect his/her status (the hard or the soft ones?), etc. We point the reader to the sketches presented in [100] for more examples of how end-users may have complex needs to satisfy.

A reasonable attempt to understand what constitutes a user-centred explanatory process in computer science is likely given by Human-Centred XAI, where proper explaining involves a conversation between the explainer and the explainee. For instance, Madumal et al. [127] formalised a model of the explanatory process using an agent dialogue framework, analysing several hundred human-human and human-agent interactions under the lens of *grounded* theory. Not surprisingly, the resulting model consists of an iterative question-answering process involving argumentation but not capturing *illocution*, considering a small range of possible explanatory contents focused on causes, justifications and processes. Similarly, also Vilone and Longo [213] proposed a conversational, argument-based explanation system for a machine-learned model to enhance its degree of explainability by employing principles and techniques from computational argumentation. Moreover, on the same line of Madumal et al., also Rebanal et al. [164] proposed and studied (only through a Wizard-of-Oz test though) an interactive approach using question-answering, to explain deterministic algorithms to non-expert users. Nonetheless, as Madumal et al. and Vilone and Longo, also Rebanal et al. focused on a small subset of possible types of explanations (i.e., `why`, `what`, `how`), avoiding *illocution*, as suggested by a few of the comments given by their user study participants: “it answers everything accurately and it gives the information that I asked for but it does so like sounding more like a glossary like a dictionary”, “... like a robot’s answers ... If I asked someone to explain it, it would not give me all this”.

3.2 Illocution as Answering to Archetypal Questions: Why Usability is a Good Metric for Explanations

Despite its compatibility, practically none of the works in XAI ever explicitly mentioned Ordinary Language Philosophy, preferring to refer to Cognitive Science [138, 90] instead. This is probably because Achinstein's illocutionary theory of explanations is seemingly difficult to implement into software by being utterly pragmatic and missing a precise definition of illocution as intended for a computer program. Therefore, in this section, we discuss how the notion of explanatory illocution can be interpreted in a more computer-friendly way.

Achinstein's theory of explanations frames the act of explaining as an illocutionary act of answering questions (cf. Section 2.2). In this sense, questions are the primary mechanism for an explainee to express her/his own needs, favouring the user-centrality of explanations. Some questions may be explicit and others not, and some may lose importance over time or vice versa. However, users are usually satisfied with explanations only when they effectively convey coverage of relevant answers for all of their goals of understanding. Though, modelling an explanatory process as a standard question-answering process gave us the first impression of being slightly unrealistic.

Think of the following example of the "university lecture": students (the explainees) follow the lessons to acquire (initially obscure) information provided by the professor (the explainer). A lesson can typically include the intervention of students in the form of observations or questions, but these interventions are, in practice, always after an initial phase of information acquisition. In other words, the professor's initial overview may not answer any preliminary questions, especially if the students know very little about what the professor is supposed to say. Regardless of this apparent lack of a question, we might all agree that the professor could still explain something good to the students.

At this point, Achinstein's theory, based on question-answering, may seem to fail to capture the need for preliminary overviews during an explanatory process, as in the "university lecture" example. Despite this first impression, we assert that overviews can also be generated as answers, partially confirming Achinstein's original theory. Indeed, for the generation of an overview, it is necessary (for the professor) to select and group information appropriately to facilitate the production of different explanatory paths

for different users (the students). We hypothesise that the way these clusters of information are created is by anticipating and answering implicit and archetypal questions, e.g., `why`, `what for`, `how`, `when`. In particular, we leverage a subtle and essential difference between "answering questions" and "explaining": *illocution*.

According to Achinstein, illocution in explaining is when "*S* utters *u* with the intention that his utterance of *u* renders *q* understandable by producing the knowledge of the proposition expressed by *u*, that it is a correct answer to *Q*" [3]. The problem with this philosophical understanding of illocution is that it is too abstract to be implementable into software, requiring one to find a way to formally frame what a *deliberate intent of explaining* is. This is why we propose a more precise denotation of explanatory illocution (for a formal definition, read Section 4.1.1).

Definition 1 (Explanatory Illocution - Informal Definition)**.** *Explaining is an illocutionary act that provides answers to an explicit question on some topic along with answers to several other implicit or unformulated questions deemed necessary for the explainee to understand the topic properly. Sometimes these implicit questions can be inferred through a thorough analysis of the explainee's background knowledge, history, and objectives, also considering Frequently Asked Questions (FAQs). However, in the most generic case, no assumption can be made about the explainee's knowledge and objectives. The only implicit questions that can then be exploited for illocution are the most generic ones, called archetypal questions.*

For example, if someone asks "How are you doing?", an answer like "I am good" would not be considered an explanation. By contrast, a different answer, such as "I am okay because I was worried I could have tested positive to COVID-19, but I am not and [...]" would generally be considered an explanation because of the intent to produce an understanding about "how you are". In other words, it answers other archetypal questions together with the main question.

We will refer to the act of explaining as *illocutionary question-answering*. Thus, we depart from Achinstein's definition, asserting that *illocution* is the primary mechanism responsible for anticipating unformulated or implicit questions (i.e., goals). This particular understanding of illocution makes Achinstein's theory practically implementable in real software applications as soon as it is possible to identify a set of archetypal questions. This is because we frame explanatory illocution as a question-answering process,

which we already know how to automate with AI (as better explained in Chapter 6).

Illocution is responsible for shaping the underlying explanatory process as *more user-centred*, helping both the explainee and the explainer consume fewer resources while communicating, thus reducing the number of explanatory steps. More precisely, we hypothesise that given an arbitrary explanatory process, increasing its ability to answer both explicit and implicit questions results in more usable explanations. In other words, the more an explanatory process is implemented as an illocutionary act of producing content-giving answers to questions, the more it is likely to meet the explanatory goals of a user and the more it will be usable.

A reasonable degree of usability is usually achieved when the (explanatory) system meets a user's specific needs. In short, we adopt the definition of *usability* as the combination of *effectiveness*, *efficiency*, and *satisfaction*, as per ISO 9241-210. ISO 9241-210 defines *usability* as the "extent to which a system, product or service can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [76]. *Effectiveness* ("accuracy and completeness with which users achieve specified goals") and *efficiency* ("resources used for the results achieved", e.g., time, human effort, costs and materials) can be assessed through objective measures. Instead, *satisfaction*, defined as "the extent to which the user's physical, cognitive and emotional responses that result from the use of a system, product or service meet the user's needs and expectations", is a subjective component, and it needs a confrontation with the user. Satisfaction is typically measured with standardised questionnaires. One of these is System Usability Scale (SUS) [34], that (despite its sometimes confusing name) is used to measure the subjective satisfaction[2] (or perceived usability) and not the usability (that according to the ISO standard is the combination of both objective and subjective metrics: effectiveness, efficiency and satisfaction) [26].

What is of utmost importance for proper user-centrality is to help the user in the process of achieving her/his own goals. So, if one agrees with Achinstein's interpretation of explanations, then in an explanatory process user's goals are identified by questions. Some questions may be explicit and others not, and some may lose importance over time or vice-versa. However, users are usually satisfied with explanations only when they effi-

[2]SUS is considered one of the most widely used standardised questionnaires for the assessment of post-test satisfaction [178, 4, 119].

ciently convey a full coverage of pertinent answers for all their objectives. Hence, since pragmatism is achieved when explanations meet the user's goal, any good explanatory tool should provide plausible mechanisms for explainees to specify their questions. Problems arise when these questions are not explicitly posed, requiring the explanatory tool to infer them automatically. It is certainly not trivial to correctly elicit the user's implicit goals, and it sometimes takes time for the user to express or understand the goals intelligibly or accurately. Sometimes these implicit questions can be deduced. However, in the most generic case, the only implicit questions that can be exploited for illocution are the archetypal ones. Specifically, when the explainee provides a precise initial question, illocution is embedded in the consequent explanation through digressions, answering other implicit questions (i.e., the archetypal ones). Instead, when the explainee gives no question but an explanandum, illocution is about providing an overview as an aggregation of different answers to implicit questions about the aspects of that explanandum, as in the example of the "university lecture" previously described.

Definition 2 (Explanandum Aspect)**.** *Given a subject to be explained or explanandum (E) under examination, an explanandum aspect (a_i; or simply aspect) denotes a distinct characteristic, feature, or element pertaining to E. Let $A = a_1, a_2, ..., a_n$ be the set of aspects associated with E. These aspects facilitate the analysis of E by partitioning it into smaller, more manageable parts. For each aspect a_i, several (archetypal) questions can be formulated, enabling a comprehensive and detailed investigation of E.*

Definition 3 (Archetypal Question)**.** *An archetypal question is an archetype applied to a specific aspect of the explanandum. Examples of archetypes are the interrogative particles (e.g.,* `why`*,* `how`*,* `what`*,* `who`*,* `when`*,* `where`*), or their derivatives (e.g.,* `why not`*,* `what for`*,* `what if`*,* `how much`*), or also more complex interrogative formulas (e.g.,* `what reason`*,* `what cause`*,* `what effect`*). Accordingly, the same archetypal question may be rewritten in several different ways, as "why" can be rewritten in "what is the reason" or "what is the cause".*

For example, if the explanandum would be "*heart diseases*", there would be many aspects involved, including "*heart*", "*stroke*", "*vessel*", "*diseases*", "*angina*", "*symptoms*". Some archetypal questions, in this case, are "`what` *is angina*" or "`why` *a stroke*".

Notably, archetypal questions prevent by design any "filling the blank" answer, thus meeting the tricky but reasonable assumption of illocution given by Achinstein for his pragmatic theory of explanations (cf. Section 2.2). What is of interest to us is that the assumption of explaining as pertinently answering (also) archetypal questions is simple and precise, removing from the "equation" the need for a model of human intention. Illocution is about anticipating the (conceivably mostly unknown) explainee's needs for an explanation by providing, as an explanation, possibly expandable summaries of (more detailed) pertinent information. In other words, the more explicit and implicit questions an explanatory process answers, the more likely the resulting explanations will meet the explainee's objectives and the more usable (effective, efficient and satisfactory) the explanatory tools. Therefore we make the following hypothesis.

Hypothesis 1 (Explanatory illocution is about answering archetypal questions)**.** *If the following premises are true:*

- *An explanatory process is an illocutionary act of providing content-giving answers to questions;*
- *Illocution is about correctly answering not just some explicit questions but also all the implicit questions that the explainee might need.*

Then, given an arbitrary explanatory process, increasing its goal-orientedness or illocutionary force results in the generation of more usable explanations. Where the goal-orientedness of an explanatory process is its ability to answer the explicit questions of an explainee, and the illocutionary force is its ability to anticipate and answer the implicit (archetypal) questions of an explainee.

To verify this hypothesis, we designed some experiments described in Part II, using the models presented in Chapter 5.

3.3 Archetypal Questions in Linguistic Theories

Casting the semantic annotations of individual propositions as narrating an archetypal question-answer pair recently gained increasing attention in computational linguistics [87, 73, 135, 160], especially in *discourse theory* and the *theory of sentential meaning representations*.

On the one hand, *discourse theory* is a branch of linguistics that studies how coherence and cohesion make up a text to form a discourse. So, discourse is said to be coherent if all its pieces belong together, while it is said to be cohesive if its elements have some common thread. In recent years, many different discourse models have been spelt out, each with different pros and cons. Amongst them, we cite the model of the *Penn Discourse TreeBank* (PDTB for short) [139, 157, 223] because it is considered one of the most generic models of discourse. In fact, with little or no change in the model, PDTB is usable for representing discourses of natural languages belonging to different families [230], e.g., Chinese, Arabic, Hindi.

The central assumption behind PDTB is that "the meaning and coherence of a discourse result partly from how its constituents relate to each other". Specifically, these relations between constituents, called discourse relations, are defined as semantic relations between abstract objects, called *Elementary Discourse Units (EDUs)*, connected by implicit or explicit connectives, e.g., "*but*", "*then*", "*for example*", "*although*". In PDTB, EDUs are spans of text denoting a single event serving as a complete and distinct unit of information that the surrounding discourse may connect to [203]. What is of interest to us is that according to Pyatkin et al. [160], all discourse connectives can be represented as questions: `in what manner, what is the reason, what is the result, after what, what is an example, while what, in what case, since when, what is contrasted with, before what, despite what, what is an alternative, unless what, instead of what, what is similar, except when, until when.`

On the other hand, the *theories of sentential meaning representation* are grammatical theories which study the relationships between predicates and arguments in a sentence. Predicate-argument relationships support answering basic questions such as "`who` did `what` to `whom`", and they can be captured with models to separate a sentence's meaning from its syntactic representation. Amongst these models, we mention the theory of *Abstract Meaning Representations (AMRs)* [15, 116], which can be used to represent whole sentences as (directed and acyclic) graphs of predicates and arguments that can be exploited for tasks such as machine translation [3], natural language generation and understanding.

[3] *Machine translation* is the conversion of sentences into symbolic knowledge representations, e.g., a piece of software written in Prolog, a logic programming language.

According to Michael et al. [135], all the AMRs can be encoded as pairs of questions and answers involving the following archetypes: `what, who, how, where, when, which, whose, why.`

Interestingly, it is possible to identify a hierarchy or taxonomy of these archetypes, ordered by their intrinsic level of generality or specificity. In particular, the simplest interrogative formulas, such as those used by AMRs, can be seen as the most generic archetypes since they consist of only one interrogative particle. Some examples of these are `what`, `why`, `when`, and `who`. We will refer to these archetypes as the *primary* ones. In contrast, the archetypes used by PDTB (e.g., `what is the reason`, `what is the result`) are more complex and specific, building over the primary archetypes. For this reason, we will refer to them as *secondary* archetypes.

Even though many more archetypes could be devised (e.g., `where to` or `who by`), we believe that the list of questions we provided earlier is already rich enough to be generally representative of any other question, whereas more specific questions can always be framed by using the interrogative particles we considered (e.g., `why`, `what`). Indeed, *primary archetypes* can be used to represent any fact and abstract meaning [27]. Instead, the *secondary archetypes* can cover all the discourse relations between them (at least according to the PDTB theory).

For example, from the sentence "*The existence and validity of a contract, or any term of a contract, shall be determined by the law which would govern it under this Regulation if the contract or term were valid*" it is possible to extract the following discourse relation about contingency (that we represent as a pair of question and answer for convenience and clarity) "*In what case would the law govern it under this Regulation? If the contract or term were valid*", and the following AMR question-answer "*By what is the existence and validity of a contract determined? The law that would govern it under this Regulation if the contract or clause were valid*". So, a discourse relation identifies two EDUs: the first encoded in the question and the second in the answer.

CHAPTER 4

Estimating Explainability: Theory and Methods

Advancing Explainable Artificial Intelligence technology requires understanding its limitations and developing better solutions through the evaluation of explainability. However, evaluation often relies on ad hoc or subjective mechanisms for measuring explainability quality as indicated in literature reviews like [212]. Explainability metrics are frequently tailored to specific XAI models [8, 171, 214, 143, 114] or depend on user studies and Cognitive Science [142, 220, 208, 37, 156]. This raises the question of whether explainability can be always objectively measured using fully automated software.

Section 2.1 emphasizes that, according to theories such as Achinstein's, not all explainable information constitutes an explanation. This implies that separate evaluation methods for explainability and explanations may be required. Usability metrics, while helpful for assessing explanations (as discussed in Section 3.2), might not be ideal for evaluating explainability due to their subjective nature and potential high costs.

Table 2.1 highlights that certain theories, such as Holland's [91] from

Cognitive Science, view explanations as perlocutionary acts. This perspective necessitates explainability evaluations to be closely tied to the user's subjective experience or outcomes. In contrast, non-perlocutionary definitions, like Achinstein's (discussed in Section 2.2), offer a more objective approach to evaluation. In fact, Achinstein's theory engages users in evaluating explanations but not necessarily explainability. This means that, according to Achinstein, information can be deemed explainable if it can be used to answer archetypal questions about the explanandum (that is, explanatory illocution, as discussed in Chapter 3), irrespective of the perlocutionary effect these answers may have.

In this section, we demonstrate how to apply Achinstein's theory to objectively evaluate explainability. We introduce the Degree of Explainability metric, an objective measure grounded in Ordinary Language Philosophy that can determine if the level of explainability is objectively poor, even if users find the resulting explanations satisfactory.

By better formalizing Definition 1 (cf. Chapter 3), we show how to quantify the degree of explainability of a set of texts. Specifically, drawing from Carnap's criteria on the adequacy of an explication (cf. Section 2.3), we define the DoX as the average explanatory illocution of information over a set of explanandum aspects. Consequently, we propose the following hypothesis:

Hypothesis 2 (DoX scores measure explainability)**.** *A DoX score can describe explainability, so that, given the same explanandum, a higher DoX implies more explainability and a lower DoX implies less explainability.*

It is important to note that the DoX metric does not assess the correctness of explanations, which is a separate aspect that should be evaluated alongside explainability. In fact, as highlighted in Section 2.3, correctness and explainability are two distinct aspects. Highly effective yet incorrect explanations can exist, and conversely, correct information may not always be easily explained.

The primary goal of the DoX metric is to identify missing explainable information, regardless of whether that information will be selected for a specific explanation. As an objective measure, it does not account for the user's subjective experience, which is a crucial aspect of the perlocutionary effects of explanations. Therefore, while the DoX metric can help identify gaps in explainable information, it does not provide insights into the effectiveness of specific explanations for individual users.

In this chapter, we will explore the theory behind DoX. Hypothesis 2 will be tested in Chapter 8. Additionally, Section 8.4 will offer a detailed discussion on the limitations of the DoX measure and potential areas for improvement.

4.1 A Formula to Quantify the Degree of Explainability

Achinstein defines the act of explaining as an act of *illocutionary question-answering*, stating that *explaining* is more than *answering a question* because it requires some form of illocution. Nonetheless, without a precise and computer-friendly definition of illocution, it is hard to go further than a philosophical and abstract understanding of such a concept. For this reason, as discussed in Chapter 3, we suggested that illocution (or, better, explanatory illocution) is in fact, the process of answering multiple generic and primitive questions (e.g., `why`, `how`, `what`) called *archetypal questions*.

For example, if someone is asking "How are you doing?", an answer like "I am good" would not be considered an explanation. Differently, the answer "I am happy because I just got a paper accepted at this important venue, and [...]" would instead be normally considered an explanation because it answers other archetypal questions together with the main question.

We are convinced that, under these premises, we can concretely measure the degree of explainability of information quantitatively. More precisely, we hypothesise that the degree of explainability of the information depends on the number of archetypal questions to which it can adequately answer. In other words, we propose to estimate the degree of explainability of a piece of information by measuring the relevance with which it can answer a (pre-defined) set of archetypal questions.

Therefore, our theoretical contribution, set out in the following subsections, consists of the precise and formal definition of: *cumulative pertinence*, *explanatory illocution*, *Degree of Explainability (DoX)*, and *average DoX*. We will first provide formal definitions and then explain them further with some examples.

4.1.1 Cumulative Pertinence, Explanatory Illocution and DoX

Assuming the correctness of a given piece of information, explainability is a property of that information. Explainability can be measured in terms of *explanatory illocution*. To understand this concept, we first introduce the definition of *cumulative pertinence*. We then provide a formal definition of explanatory illocution and present DoX. Lastly, we discuss the need for an average DoX metric for comparing explainability between different systems.

Definition 4 (Cumulative Pertinence)**.** *The cumulative pertinence is an estimate of how pertinently and how in detail a given piece of information* Φ *can answer a question about an explanandum aspect* $a \in A$ *(cf. Definition 2). Let* D_a *be the subset of all the details (e.g., sentences, grammatical clauses*[1]*, paragraphs) in* Φ *that are about an aspect* a*. Let* q_a *be a question about* a*. Let* $p\left(d, q_a\right) \in [0, 1]$ *be the pertinence of a detail* $d \in D_a$ *to* q_a*. Let also* t *be a pertinence threshold in the* $[0, 1]$ *range. Then, the cumulative pertinence of* D_a *to* q_a *is* $P_{D_a, q_a} = \sum_{d \in D_a, p(d, q_a) \geq t} p\left(d, q_a\right)$.

A *pertinence threshold*, in the context of the cumulative pertinence definition, represents a predefined level of relevance or significance that a detail must possess to be considered while estimating the cumulative pertinence of an information piece Φ. The threshold ranges from 0 to 1, with 0 indicating no relevance and 1 representing complete relevance.

The pertinence threshold is crucial for several reasons. First, it serves as a filter, helping to eliminate information that is not relevant or significant enough to answer a specific question about an explanandum aspect. By setting a threshold, only details with a pertinence value equal to or higher than the threshold are considered in the cumulative pertinence calculation. Second, the threshold enables a more precise estimation of the cumulative pertinence. By excluding details with pertinence values below the threshold, the cumulative pertinence becomes a better representation of how well the information piece can answer a question about a specific explanandum aspect.

Moreover, the pertinence threshold encourages focus on the most pertinent details when assessing the explanatory quality of a given piece of

[1] A typical *clause* consists of a subject and a syntactic predicate, the latter typically a verb phrase composed of a verb with any objects and other modifiers.

information. This is especially important when dealing with large amounts of data, as it helps DoX concentrate on the most relevant details for the evaluation of explainability. Lastly, the pertinence threshold offers flexibility, as it can be adjusted according to specific needs, allowing for customized evaluations of explainability. For instance, a lower threshold might be used when a broader understanding of an explanandum aspect is desired, while a higher threshold would be more suitable when only highly pertinent details are of interest.

Building on the definition of cumulative pertinence, we can now provide a formal definition of explanatory illocution.

Definition 5 (Explanatory Illocution - Formal Definition)**.** *The explanatory illocution is a set of cumulative pertinences for a pre-defined set of archetypal questions. Let Q be a set of archetypes q and q_a be the question obtained by applying the archetype q to an aspect $a \in A$. Then the explanatory illocution of Φ to an aspect $a \in A$ is the set of tuples $\{\forall q \in Q| < q, P_{D_a,q_a} >\}$*[2].

Consequently, we define DoX as follows.

Definition 6 (Degree of Explainability)**.** *DoX is the average explanatory illocution per archetype, on the whole set A of relevant aspects to be explained. In other terms, let $R_{D,q,A} = \frac{\sum_{a \in A} P_{D_a,q_a}}{|A|}$ be the average cumulative pertinence of D to q and A, where $D = \{\forall a \in A, \forall d \in D_a | d\}$, then the DoX is the set $\{\forall q \in Q| < q, R_{D,q,A} >\}$.*

However, DoX alone cannot help in judging whether some collections of information have higher degrees of explainability than others. This is because DoX is a set, and sets are not sortable. Thus we combine the set of pertinence scores composing DoX into a single score representing explainability, called *average DoX*. So, the resulting *average DoX* can act as a metric to judge whether the explainability of a system is greater than, equal to, or lower than another.

Definition 7 (Average Degree of Explainability)**.** *The Average DoX is the average of the pertinence of each archetype composing the DoX. In other terms, the Average DoX is $\frac{\sum_{q \in Q} R_{D,q,A}}{|Q|}$.*

[2]The operator $< x, y >$ is used here to represent tuples.

The average DoX represents a naive approach to quantify explainability with a single score, as it implies that all the archetypal questions and aspects have the same weight. However, this may not necessarily be true. As suggested by Liao et al. [124], it seems that there is a shared understanding that `why` explanations are the most important in XAI, sometimes followed by `how`, `what for`, `what if` and, possibly, `what`. In other words, the relevance of an explanation can be estimated by the ability to effectively answer the most relevant (archetypal) questions for the stakeholders' objectives. Nonetheless, defining which (archetypal) question is the most relevant is challenging and somewhat subjective. Therefore we believe that average DoX is probably the only objective solution to this dispute.

We will now discuss some examples of applying the formulas mentioned earlier. We will also demonstrate how these formulas can measure Carnap's adequacy criteria.

4.1.2 Interpreting DoX in Terms of Carnap's Criteria

Suppose the sentence "I am happy that my article has been accepted in this prestigious journal" is given as Φ and the set of relevant aspects *{heart, stroke, vessel, disease, angina, symptom}* as A. In this case, the set of details D contains the following details:

- "I am happy";
- "my article has been accepted in this prestigious journal";
- "I am happy that my article has been accepted".

However, none of the details above is about the explanandum. Thus $D_a = \emptyset, \forall a \in A$, because nothing in Φ is related to A. Hence, the average cumulative pertinence would be equal to 0 for every archetype $q \in Q$, forcing the DoX score to be equal to 0, as expected. In other words, no detail of Φ would explain anything about A. Therefore the explainability of Φ for A would be zero.

On the contrary, we would not have a null DoX for A when using the sentence "angina happens when some part of your heart does not get enough oxygen" as Φ. That is because the new Φ contains details about at least two relevant aspects in A: "angina" and "heart". Such details would

score a higher average cumulative pertinence $R_{D,q,A}$ for q equal to `why` because they are about causality.

Eventually, when computing the DoX of the new Φ for this set of explanandum aspects A with the DoXpy algorithm presented in Chapter 8[3], the average DoX is 0.29. In particular, as expected, the archetypes with the best score are the ones related to causality (i.e., `what effect` has a score of 0.59; `in what case`, `why` and `how` have a score of 0.57). In contrast, many of the other archetypes have a null score (i.e., `who`, `when`).

Given Definition 6, we can say that DoX is an estimate of the *fruitfulness* of D that combines in one single score the *similarity* of D to A and the *exactness* of D for Q. For these reasons, DoX is akin to Carnap's central criteria of adequacy of explanation (introduced in Section 2.3). Although, differently from Carnap, our understanding of exactness is not that of adherence to standards of formal concept formation[4] [36], but rather that of being precise or pertinent enough as an answer to a given question.

The number of relevant explanandum aspects covered by a given piece of information, and the number of details that are pertinent about it (i.e., $|\{\forall a \in A, \forall d \in D_a | d\}|$), roughly say how much *similar* that information is to the explanandum. More precisely, the formula used for computing the cumulative pertinence P_{D_a,q_a} sums the contribution of every single detail according to its pertinence to the aspects $a \in A$, telling us how much D_a is similar to a. Thus, if pertinence $p\,(d, q_a)$ would close to zero for all archetypes $q \in Q$, then a detail d would have nothing to do with an aspect a. Furthermore, the average cumulative pertinence $R_{D,q,A}$ contains information about the *exactness* of multiple answers, aggregating pertinence scores. As a result, by measuring $R_{D,q,A}$ for all the $q \in Q$, we also obtain an estimate of how D is *fruitful* for the formulation of many other different explanations intended as the result of an illocutionary act of pragmatically answering questions.

This construction of DoX, according to Carnap's main adequacy criteria and according to the interpretation of explanatory illocution presented in Chapter 3, is crucial because it allows for the implementation of an algorithm for quantifying explainability, as discussed in Chapter 8.

Although DoX cannot be employed directly on black-box models, it

[3] When using the MiniLM pertinence estimator introduced in Section 8.1.

[4] Actually, Carnap did not specify what he means by "exactness". Regardless, in this context, "exactness" is often viewed as either lack of vagueness or adherence to standards of formal concept formation.

can be applied to the output of XAI algorithms or to any other explainable information to understand how that information can be used to explain. In this sense, DoX is the most useful when used to evaluate extensive collections of explainable information (e.g., the output of an ensemble of XAI algorithms).

4.2 Legal Compliance: a Comparison of DoX with other Explainability Metrics

In Section 1.3, we clarified the existence of explainability obligations and their extent. So, in this section, we use these obligations to compare DoX with other explainability metrics and understand to which extent these metrics could be used for assessing explainability in compliance with the law.

Measuring the degree of explainability of AI systems has become relevant in the light of research progress in the XAI field, the proposal for a European Regulation on Artificial Intelligence, and ongoing standardisation initiatives that will translate these technological advancements in a *de facto* regulatory standard for AI systems. To date, standardisation entities have proposed white papers and preliminary documents showing their progress[6], among them we mention:

- The European Telecommunications Standards Institute[7]: "[w]hen it comes to AI capabilities as part of new standards, there is a need to revise these models, by identifying appropriate reference points, AI sub-functions, levels of explicability of AI, quality metrics in the areas of human-machine and machine-machine interfaces, etc."

The work presented in Section 4.2 and Section 1.3 was developed in collaboration with Salvatore Sapienza from the University of Bologna [198]. *S. Sapienza*: legal analysis constituting Section 1.3, part of the introduction of [198]. *S. Sapienza* and *F. Sovrano*: definition of the four main principles for explainability metrics introduced in Section 4.2. *F. Sovrano*: the remaining part of [198], including the analysis of philosophical theories of explanation, all the tables and the analysis of explainability metrics.

[5]This table extends a similar one in [198].

[6]An extensive list of examples is available at `https://joinup.ec.europa.eu/collection/rolling-plan-ict-standardisation/artificial-intelligence`

[7]`https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp34_Artificial_Intellignce_and_future_directions_for_ETSI.pdf`, p. 23

Table 4.1: ***Comparison of different explainability metrics***[5]**.** *Column "Source" points to referenced papers while "Metrics" to the names of the metrics mentioned in the papers. The remaining columns are the explainability dimension discussed in Section 4.2. Elements in bold are the best column-wise.*

Source	Model Information Format	Closest Supporting Theory	Subject based	Measured Carnap's Criteria	Metrics
[171]	Rule-based	Causal Realism	**No**	Exactness, Fruitfulness	Performance Difference, Number of Rules, Number of Features, Stability
[214]	Rule-based	Causal Realism	**No**	Similarity, Fruitfulness	Fidelity, Completeness
[143]	Feature Attribution	Causal Realism	**No**	Exactness, Fruitfulness	Monotonicity, Non-sensitivity, Effective Complexity
[114]	Rule-based	Causal Realism	**No**	**Similarity, Exactness, Fruitfulness**	Fidelity, Unambiguity, Interpretability, Interactivity
[92]	**Any**	Causal Realism, Cognitive Science, Naturalism & Co.	Yes	Exactness, Fruitfulness	System Causability Scale
[90]	**Any**	Cognitive Science, Naturalism & Co.	Yes	Exactness, Fruitfulness	Satisfaction, Trust, Mental Models, Curiosity, Performance
[63, 142] [220, 208] [156, 37]	**Any**	Cognitive Science, Naturalism & Co.	Yes	Exactness, Fruitfulness	Usability: Effectiveness, Efficiency, Satisfaction
[8]	Heatmap	Constructive Empiricism	**No**	Similarity, Exactness	Relevance Mass Accuracy, Relevance Rank Accuracy
[143]	Prototype-based	Constructive Empiricism	**No**	Similarity, Fruitfulness	Non-Representativeness, Diversity
DoX	**Any** (Natural Language Text)	Ordinary Language Philosophy	**No**	**Similarity, Exactness, Fruitfulness**	Degree of Explainability

- The CEN-CENELEC[8], proposing to "[d]evelop research-based metrics for explainability (to tie in with high-level conceptual requirements), which can be developed into pre-standards like workshop agreements or technical reports".

- ISO/IEC TR 24028:2020(E), stating that "[i]t is important also to consider the measurement of the quality of explanations", providing for details on the key measurements, i.e., continuity, consistency, selectivity.

Let us remind the reader that, under the proposed AI Act, adopting a standard means certifying the degree of explainability of a given AI system. Therefore, metrics become helpful in the course of the standardisation process: *i) ex-ante*, when defining the explainability measures adopted by the standard; *ii) ex-post*, when verifying in practice the adoption of a standard. From these premises, it follows that in the light of the purposes of the AI Act set out in Section 1.3, any explainability metric should respect at minimum the following **main principles**, by being:

- **Risk-focused**: this means that the metric should be functional to measure the extent to which the explanations provided by the system allow for an assessment of the risks to the fundamental rights and freedoms of the persons affected by the system's output. This is necessary to ensure user-enabling (e.g., Art. 29) and compliance-oriented (Annex IV) explainability.

- **Model-agnostic**: this means that the metric should be appropriate to all the AI systems regulated by the proposed AI Act[9].

- **Flexible & Goal-aware**: this means that the metric should be flexible to the different needs of potential explainees (e.g., AI system providers and users, standardisation entities)[10] and applicable to all high-risk AI applications listed in Annex III of the Act.

[8] `https://ftp.cencenelec.eu/EN/News/PolicyOpinions/2020/CEN-CLC_AI_FG_White-Paper-Response_Final-Version_June-2020.pdf`, p.8

[9] Annex I provides a list of the AI techniques and approaches that fall within the remit of the Regulation.

[10] Since it might be hard to determine *ex-ante* the nature, the purpose, and the expertise of the explainee, the metrics should consider the highest possible number of potential explainees.

- **Intelligible & accessible**: this means that if the information on the metrics is not accessible (e.g., due to intellectual property reasons) or the results of a metric are not reproducible (e.g., due to a subjective evaluation), explainees will confront with a situation of uncertainty, as an *ignotum per ignotius*. This would contradict the risk minimisation principle.

We now identify some pros and cons of existing metrics (and measures) to quantitatively estimate the degree of explainability of information to understand their range of applicability across different needs and interpretations of explainability. We perform a qualitative classification of these measures based on Carnap's desiderata, the theories of explanation presented in Section 2.1 and the aforementioned main principles.

More specifically, in Table 4.1 we rank explainability metrics on the following dimensions:

- *Information Format*: the information format supported by the metric, e.g., rule-based, example-based, natural language text.
- *Supporting Theory*: the supporting philosophical theory of the metric, e.g., Cognitive Science, Constructive Empiricism.
- *Subjectivity*: whether the metric requires evaluations given by human subjects.
- *Covered Adequacy Criteria*: the adequacy desiderata (see Section 2.3) measured by the metric, i.e., similarity, exactness, fruitfulness.

Then, in Table 4.2 we align the supporting theories (thus also the metrics) to the properties identified with the analysis of the AI Act.

Differently from ISO/IEC TR 24028:2020(E) we do not focus on metrics specific to ex-post feature attribution explanations, so we selected methods possibly applicable also to ex-ante or more generic types of explanations. Feature Attribution is only for explainability about causality, hence being more centred on Causal Realism, while our investigation tries to compare different metrics across the supporting philosophical theories.

As shown in Table 4.1, we were able to find at least one example of metric for each supporting philosophical theory, with a majority of metrics focused on Causal Realism. Notably, in all the metrics supporting Causal

Realism, measurements of exactness and fruitfulness are coincident. Common to all the metrics based on Cognitive Science and Naturalism/Scientific Realism is that they require human subjects to perform measurements. Therefore they tend to be more expensive than the others, at least in terms of human effort. Furthermore, those proposing heuristics to measure all Carnap's desiderata are just two, one for Causal Realism [114] and the other is DoX. Interestingly, Lakkaraju et al. [114] evaluate the three desiderata separately, while DoX is the only known metric combining all of them in a single score.

Table 4.2: ***Alignment of explainability definitions to explainability properties from the AI Act.*** *Every row stands for a different theory of explanations and, therefore, for a different explainability definition. The considered theories of explanations and definitions of explainability are discussed in Section 2.1 and Table 2.1.*

	Risk-Focused	**Model-Agnostic**	**Flexible & Goal-Aware**	**Intelligible & Accessible**
Causal Realism	Yes, if understanding risks implies understanding causality	Not available yet	No, it's not pragmatic and it considers only goals related to causality	Yes, it can be
Constructive Empiricism	Yes, if explaining risks is about answering `why` questions	Not available yet	No, it focuses only on `why` questions	Yes, it can be
Ordinary Language Philosophy	Yes, it can be	Yes, if all explanations can be represented with natural language	Yes	Yes, it can be
Cognitive Science	Yes, it can be	Yes, the evaluation is subject-based	Yes	Unlikely. All the subject-based metrics may be very expensive and hard to reproduce, this makes them less accessible
Naturalism and Scientific Realism	Yes, it can be	Yes, the evaluation is subject-based	Yes	Unlikely. It relies on (usually) expensive subject-based metrics

Let us now discuss the extent to which philosophy-oriented metrics measure explainability and can match the requirements set by the proposed AI Act. First, under the AI Act, metrics should allow the measurement of the capability of the system to provide information related to the risks posed to fundamental rights and freedoms of the persons affected by the

system. This is a form of *goal-aware* explainability, thus calling for a pragmatic interpretation of explanations as that of all the theories identified in Section 2.1, but not Causal Realism. Then, metrics shall be appropriate to the list of AI approaches listed in Annex I. This entails that only those based on a *model-agnostic* approach to explainability can ease compliance to the proposed AI Act unless a combination of different model-specific metrics is envisaged. The results shown in Table 4.2 indicate that the metrics supported by both Causal Realism and Constructive Empiricism might struggle at being model-agnostic and goal-aware. This probably limits their applicability to particular contexts.

Furthermore, metrics shall also be adaptive to the several market sectors which can observe a substantial deployment of high-risk AI systems. Therefore, given the horizontal application of the AI Act and the contextual applicability of sectoral legal frameworks, we have that any explainability metric should be *flexible* enough to adapt to different technological constraints and explanation objectives. Considering that explanation objectives can be framed in terms of questions to answers, definitions of explanations that are focused solely on specific enquiries, such as Causal Realism and Constructive Empiricism, may struggle to meet the adaptivity requirement. Moreover, flexibility towards all the potential explainees entails that all subject-based metrics (i.e., those inspired by Cognitive Science or Naturalism/Scientific Realism) require many explainees to be tested and standardised.

Finally, the *intelligibility, accessibility and understandability* of metrics require a metric to also be economically accessible. However, all the subject-based metrics may be expensive, thus making the metric less accessible to some. The same goes with intelligibility and those metrics developed under standards that are not open to public scrutiny.

It follows that those metrics based on Achinstein's theory of explanations (i.e., DoX) are more likely to align with the explainability requirements of the proposed AI Act. This is because they would not rely on a perlocutionary understanding of explanations (i.e., they would not be subject-based) and would be flexible enough to consider evaluations of explainability beyond causality.

CHAPTER 5

Explanatory Artificial Intelligence: Theoretical Foundations

In Chapter 2, we identified different definitions of explanations and explainability. In Chapter 1, we also identified existing legal and ethical requirements that some explanations should possess. So we hereby provide the theoretical foundations of how to automatically generate user-centred explanations or YAI for short. To do so, we discuss one-size-fits-all explanations and why they are insufficient for user-centrality. Then we draw the difference between XAI and YAI, providing a formal definition of *user-driven explanatory tool* and *explanatory space*. Soon after, we discuss the main properties of an explanatory space and some heuristics to explore it in a user-centred and efficient way. Therefore, the purpose of this chapter is to build on the growing awareness that good explanations start from, but are not, the output of improved form of XAI, but constitute a complementary and vastly different endeavour.

5.1 User-Centrality and the Problem with One-Size-Fits-All Explanations

Computational irreducibility is typical of emerging phenomena such as physical, biological and social ones [17]. For these systems, it is possible to simulate every step of the system's behaviour evolution. However, it is only possible to predict this simulation's result by letting the system take each evolutionary step. Thus, standing on the definition given by Zwirn and Delahaye [231], user-centrality in explaining to humans is *computationally irreducible* because generally speaking, nothing besides the user itself (while unfolding an explanation) can predict whether an explanation is beneficial, usable or satisfactory.

Therefore, we take a strong stand against the idea that static, one-size-fits-all approaches to explanations have a chance of being pragmatic (i.e., user-centred). This is to say that XAI-based tools answering just a few specific `why`, `how` and `what` questions are not enough for properly explaining in a user-centred way. One-size-fits-all explanations are based on the idea that the same piece of information can fit all, therefore assuming that it would be usable and valuable *a priori* for anybody.

The **main types of one-size-fits-all explanations** are the following:

- **Normal XAI-based explanations**: answering one only question or just few.
- **Selected narratives**: answering only one type of question, e.g., how-why narratives answering only `how` or `why` questions.
- **Overwhelming explanatory closures**: explaining by giving large dumps of explainable information. A *1st-level explanatory closure* is about immediately presenting all the available information. A *2nd-level explanatory closure* is about providing such information in two rounds. *3rd-level explanatory closures* are like 2nd-level ones, but all the information is provided after two levels of interaction, and so on.

Indeed, a one-size-fits-all explanation, to fit all, should contain all the possible answers to all the possible questions of all the possible users (e.g., an overwhelming explanatory closure). However, this type of explanation would become useless for a human, being overwhelming in size and content, as soon as the complexity of the explanandum increases beyond a relatively trivial threshold. In other words, an explainable dataset or system,

per se, is not a user-centred explanation, whereas user-centrality requires a generic *nth-level explanatory closure*.

A user's interest in the output of an explanation system often lies in a few short statements out of the hundreds of thousands that the explanation system can generate. These few lines depend on the function the user gives to the explanation. Hence we must assume that, in general, the purpose of explanations is known to the user rather than to the explanation system, and it cannot be decided in advance. However, it becomes knowable only during the evolution of the task for which the explanation is required.

For example, a complex big-enough *explainable* software can be difficult to *explain*, even for an expert, and the optimal (or even sufficient) explanation might change from expert to expert. In this specific example, explainable software is necessary, but more is needed for explaining.

5.2 XAI vs YAI

A user-centred explanatory tool requires providing goal-oriented explanations. Goal-oriented explanations imply explaining facts relevant to the user, according to her/his background knowledge, interests and other peculiarities that make him/her a unique entity with unique needs that may change over time. Therefore, to model a user-centred explanatory process, we need to:

- Disentangle *making things explainable* (i.e., XAI) from *explaining* (i.e., YAI): in a way, this is tantamount to separating the presentation logic from the application logic. Only explaining has to be user-centred. In this sense, we like to say that we need both the Xs and the Ys of AI[1].

- Design a *presentation logic* that allows personalised explanations out of some explainable information.

In Figure 5.1, we show a simple model of an explanatory tool, which separates between explainability and explanations. In order to increase the overall cohesion of the explanatory system, in this model, we require an explicit logical separation between the functionalities related to *producing explainable information* and those related to *producing pragmatic explanations*. In addition, we envision another logical separation in the production

[1]XX and XY are the human chromosomes responsible for biological gender.

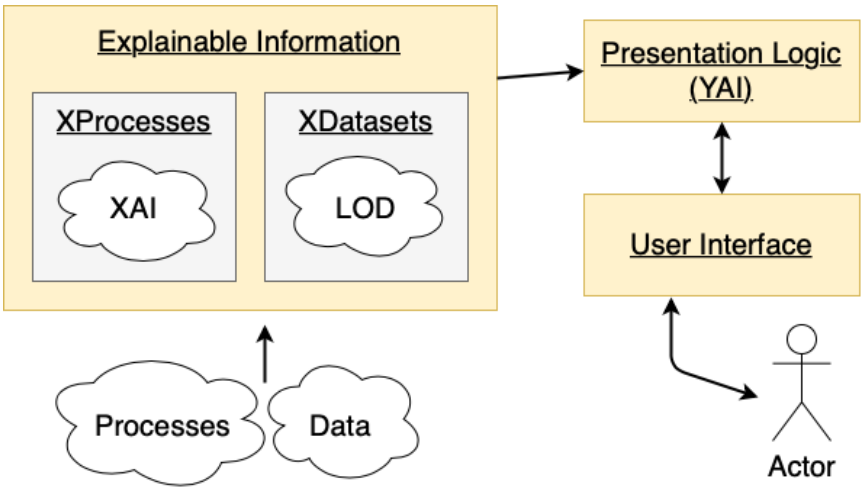


Figure 5.1: ***XAI vs YAI: an abstract model of explanatory tool****. The model depicted in this figure shows how to decompose the flow of explanatory information that moves from raw representations of processes/data to the explainee (or actor). Raw data are refined into explainable datasets (e.g., Linked Open Data, LOD for short). Raw processes are refined into explainable processes. Explainable information can be used by YAI to generate practical explanations.*

of actual explanations between *building explanations* (i.e., the presentation logic) and *interfacing with users*. Independently, *producing explainable information* should be separated in *generating explainable processes* and *producing explainable datasets*.

One of the most valuable benefits coming from this distinction of YAI from XAI is that it would meet the Single Responsibility Principle[2] [129], making easier to integrate an explanatory layer in an existing application layer (without changing the latter). We can see that nowadays, the presentation logic is not explicitly separated from the application logic in many XAI applications intended as explanatory tools.

[2]The *principle of single responsibility* is a computer programming principle that states that each module or class must have responsibility for a single part of the functionality provided by the software and that the class must entirely encapsulate this responsibility.

5.3 Definition of User-Centred Explanatory Process and Space: the SAGE Properties

We believe that an explanatory tool is an instrument for articulating explainable information into an explanatory discourse. This definition of an explanatory tool is drawn from the essential **best practices of scientific inquiry** [21], involving:

- **Sense-making of phenomena**: classical question-answering to collect enough information for understanding, thus building an explainable explanandum (perhaps through XAI).
- **Articulating understandings into discourses**: re-ordering and aggregation of explainable information to form an explanatory narrative or, more generally, a discourse to answer research questions.
- **Evaluating**: pose and answer questions about the quality of the presented information (e.g., argue them in a public debate).

More formally, we propose the following definition of the explanatory process, considering that for user-centrality, an explainee must be able to specify as input of the process her/his goals, otherwise not inferable due to the computational irreducibility of the phenomenon.

Definition 8 (Explanatory Process)**.** *Let an explanans (plural is explanantia) be a text in natural language (i.e., English) answering one or more questions. A user-driven explanatory process or explanatory discourse articulation (stylised in Figure 5.2) is a function p for which $p\left(D, E_t, i_t\right) = E_{t+1}$, where:*

- *D is the explanandum: a set of explainable pieces of information; a set of answers organised to build archetypal explanations that are useful to the explainee.*
- *E_t is the explanans, at time step $t \geq 0$. E_t can be any meaningful rephrasing of the information in D.*
- *i_t is the interaction of the explainee at step t.*

We can iteratively apply p, starting from an initial explanans E_0, until satisfaction. The user interaction i is a tuple made of an action a taken from

the set A_p of possible actions for p and a set of auxiliary inputs required by the action a. Whenever A_p allows any explainee to specify its needs and goals to maximise the usability of E_{t+1}, then p is said to be user-centred.

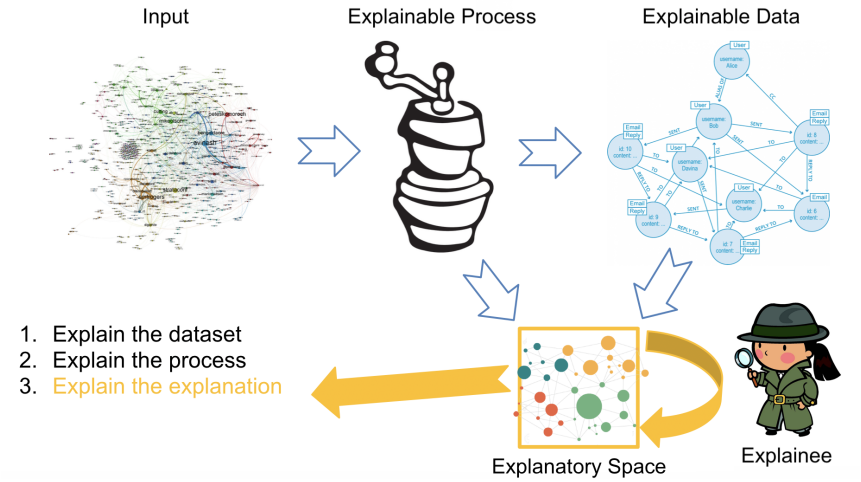


Figure 5.2: ***Stylized interactive explanatory process.*** *A user-centred explanatory process explains an explanandum to an explainee, thus producing as output an explanans that is meaningful for the specific explainee.*

To understand how to implement such a user-centred explanatory process p, we need first to define the characteristics of the space of all the possible explanations generated by p. We call this space of explanations the *explanatory sub-space* of p.

Definition 9 (Explanatory Sub-Space)**.** *An explanatory sub-space is a hypergraph $H_p = (\xi_p, \epsilon_p)$ of interconnected explanantia reachable by an explainee interacting with a process p, given an explanandum D, a set of actions A_p and an initial explanans E_0. Thus, the set of hyperedges ϵ_p is the set of all possible explanantia that can be generated by p about D:*

$$\epsilon_p = \{E_0\} \cup \{\forall u > 0, \forall i_u \in A_p \mid p\,(D, i_u, E_{u-1})\}$$

While the set of nodes ξ_p is the set of questions q and answers a covered by the explanantia[3]*:*

$$\xi_p = \{\forall E \in \epsilon_p, \forall < q, a > \in E \mid q\} \cup \{\forall E \in \epsilon, \forall < q, a > \in E \mid a\}$$

[3]It is always possible to represent natural language sentences as networks of questions and

This leads us to the definition of an *explanatory space*.

Definition 10 (Explanatory Space)**.** *An explanatory space is the hypergraph $H = (\xi, \epsilon)$ resulting from the union of the explanatory sub-spaces of each p in the set of all the possible explanatory processes P. In particular, we have that:*

$$\epsilon = \{\forall p \in P, \forall E \in \epsilon_p \,|E\}$$
$$\xi = \{\forall p \in P, \forall i \in \xi_p \,|i\}$$

Therefore, according to Definition 8, we have that an explanatory subspace H_p, in order to be user-centred, should be *adaptable*. More specifically, it should be:

- ***S**ourced*: bound by the explanandum D. The space should be a description of D.
- ***A**daptable*: bound by the narrative purposes of the explainee and his/her queries i. The space should be structured to minimise the number of queries for the explainee to achieve its objective.
- ***G**rounded*: bound by the explanatory process p as illocutionary question-answering. The space should be structured in order to effectively and efficiently answer questions.
- ***E**xpandable*: bound by the characteristics of the web of explanantia E. The space should form a coherent information network that can be explored and described through linguistic structures such as narration or, more generally, discourse.

We will refer to these **properties of an explanatory sub-space** as the *SAGE properties*, and we will use them to define a set of actions A_p to embed user-centrality in an explanatory process.

5.4 Efficient Exploration of Explanatory Spaces: the ARS Heuristics

In graph theory, tree decompositions are used to speed up solving some computational issues on graphs (and more generally hypergraphs) [81]. Indeed, many NP-difficult problems on graphs can be efficiently solved via

answers. Indeed, casting the semantic annotations of individual propositions as narrating a question-answer pair recently gained increasing attention in computational linguistics [87, 73, 135, 160].

tree decomposition [13]. So, suppose an explanatory space is a hypergraph. In that case, any efficient explanatory process p should be able to approximate a decomposition of such hypergraph into some hypertree, allowing the explainees to efficiently navigate through the vast underlying space and find the answers they are seeking.

More specifically, decomposing an explanatory space H into a hypertree is equivalent to ordering and prioritising all the explanantia and the pieces of information within the explanantia. So that the explainee can efficiently navigate and read the explanatory space from the root (i.e., any initial explanans) to the leaves of its decomposition as a sequence of information. Though, several different hypertree decompositions might exist for the same explanatory space with no assurance that all of them are effective as they should be at explaining to a human. That is because the output of an explanatory process should be pragmatic and user-centred. A good explanatory process should be able to adapt to the needs of a human explainee with specific background knowledge and specific goals.

To this end, we propose a few **heuristics for user-driven exploration of an explanatory space**, designed to maximise the *adaptability* of the explanatory process. These heuristics are namely:

- ***A**bstraction*: used to identify the nodes (also called *explanandum aspects*) of the decomposition of the explanatory space. This is done by aggregating explanations according to some taxonomy defining a hierarchy of abstractions.

- ***R**elevance*: used to order the information about explanandum aspects according to its relevance (e.g., to the explainee's objectives).

- ***S**implicity*: used to select the viable edges of the decomposition and information about an explanandum aspect. This can be done by filtering the content of the explanandum aspects or also by prioritising specific abstractions over others.

We will refer to these heuristics as the *ARS*[4] *heuristics*.

By definition, both the SAGE properties and the ARS heuristics (the SAGE-ARS model) pose some constraints on the ways of interaction that allow exploring the explanatory space. In other words, these constraints help to define a set A_p of actions that would allow the user to explore in

[4]"Ars" means art in Latin.

a user-centred way a decomposition of the whole explanatory space starting from an initial explanans E_0 (i.e., the output of a XAI), according to Definition 8.

Following the definition of explanatory tool drawn from the best practices of scientific inquiry described in Section 5.3, some **primitive actions** that can be implemented are:

- **Open-ended question-answering**, for *sense-making of phenomena*: the user writes a question and gets one or more relevant punctual answers.
- **Aspect overviewing**, for *articulating understandings*: the user selects an aspect of the explanandum receiving as explanation a set of *relevant* answers to archetypal questions about that aspect or related aspects. The user can control the number of answers and questions composing the overview by increasing or decreasing the level of detail following the *simplicity* heuristic.
- **Argumentation**, for *evaluating*: the user evaluates the explanations, identifying counter-arguments or weak points that can be used for further (automated) reasoning.

The first two primitive actions are said to be the main primitives for explaining because they align to *sense-making* and *articulation of understandings*. We previously defined an *explanatory tool* as "an instrument for articulating explainable information".

Specifically, an *overview* is an appropriate summary of an explanandum aspect. In contrast, a *specific answer to a question* can be seen as a sequence of information (a path) that can span more explanandum aspects. Thus, for each SAGE property, we can identify a set of **SAGE commands** that implement these primitive actions the explainee can use during the explanation process:

- *"**S**ourcing" commands*: used to access the source of an explanation fragment (e.g., a law, a scholarly paper, a rule).
- *"**A**dapting" commands*: used to provide the explanatory process with sufficient information to model the background knowledge and the goals of the explainee in order to personalise the content of the explanations.

- *"**G**rounding" commands*: used to ask questions.
- *"**E**xpanding" commands*: used to navigate the decomposition of the *explanatory space* and get a partial view of it. Examples of expanding commands might be:
  - *Get Overview*: it opens an explanatory overview of a concept.
  - *More*: it shows additional details available in the explanation but currently hidden from the interface because of the simplicity policy.
  - *Less*: it removes the information added with the "More" and "Get Overview" commands.

In the following section, we will show a "proof of concept" YAI providing concrete examples of the above commands. Alternatively, more concrete instances of the SAGE-ARS model are YAI4Hu and YAI4Edu, two explanatory tools approximating an nth-level explanatory closure, which are described in Chapter 6 and Chapter 10.

In particular, we make the following hypothesis, tested with experiments described in Chapter 7.

Hypothesis 3 (The SAGE-ARS model produces user-centred explanations)**.** *An explanatory process that sufficiently implements the ARS heuristics and the SAGE commands is more user-centred than any one-size-fits-all explainer, producing better explanations through an easy-to-navigate decomposition of the explanatory space. In other terms, not all the decompositions of an explanatory space are equally useful to explainees if no assumption is made about their background knowledge.*

5.5 Proof of Concept: a YAI compliant with the GDPR

We hereby present a proof of concept[5] of the SAGE-ARS model applied to a YAI for explaining an automated decision-making system in a real-case scenario where explanations compliant with the GDPR are required (as described in Section 1.1).

The real-case scenario concerns the conditions applicable to minors' consent to online information services. Article 8 of the GDPR sets the

[5]Real software implementations are discussed in Part II.

minimum age for consent without parental authorisation at 16. Domestic law could derogate this limit, and in Italy, legislative decree 101/2018 sets it at 14 years. In order to automate the verification of the proper application of existing rules, an online information service encodes the fact that Art. 8 of the GDPR is superseded by the Italian legislative decree 101/2018 with LegalRuleML [11, 152], representing the rules with defeasible logic [146]. Moreover, the online information service uses the legal reasoner SPINDle [115], a logic-based AI, to process the correct rules according to jurisdiction (e.g., Italy) and age.

Suppose that Marco (a 14 years old Italian teenager living in Italy) uses Whatsapp, and his father, Giulio, wants to remove Marco's subscription to Whatsapp without Marco's consent because he is concerned about Marco's privacy when online. In this simple scenario, the automated decision-making system based on SPINDle would reject Giulio's request to remove Marco's profile because of the Italian legislative decree 101/2018. What if Giulio wants an explanation of the automated decision? The online information system would have to use an explanatory process.

According to the definition given in Section 5.3, a user-centred (interactive) explanatory process is about explaining an explanandum to an explainee (reader and narrator), thus producing as output an explanans that is meaningful for the specific explainee. In this scenario, the explanandum consists of the following:

- A *rule-base* serialised in LegalRuleML;
- A *dataset of premises* (the information about involved entities);
- A *dataset of conclusions* obtained by applying the premises to the rule base;
- A *causal chain* (i.e., the ordered chain of decisions taken by SPINDle to produce the conclusions).

Assuming that the explanandum is provided as defined above, then: How do we define the explanatory process? How do we pick the initial explanans? How do we pick the set of actions? To answer these questions, we use the SAGE-ARS model.

The *simplicity* heuristic (the S of ARS) implies that the explanatory process should start from a very minimal and simple explanans, adding information iteratively (and interactively) in a stepwise manner. Simplicity

also implies that simple representations of the explanandum (e.g., natural language descriptions) should be presented before the original representations (e.g., the XML representation of the LegalRuleML encodings). Notably, the simplicity heuristic is mentioned in recommendation 29.5 of the "Policy and Investment Recommendations" of the AI-HLEG [148]: "Ensure that the use of AI systems that entail interaction with end-users is by default accompanied by procedures to support users [...]. These procedures should be accompanied by simple explanations and a user-friendly procedure". Physical constraints limit the amount of information that can be effectively provided to a human. Therefore, simpler explanations are more likely to be accepted and understood and tend to be better than complicated ones. Thus they should be presented earlier.

Furthermore, the heuristic of *abstraction* (the A of ARS) implies that the explanans must be organised in such a way as to help the explainee move from abstract and generic information to more specific and factual information, or vice versa. The abstraction policy is motivated by the assumption that the possibility of isolating and precisely defining aspects of the explanandum is advantageous, as it facilitates the focusing of attention.

Finally, the *relevance* heuristic (the R of ARS) implies that the initial explanans should contain the most relevant information possible and that further details should first concern the entities directly involved in the initial explanans. Instead, other entities should be explored/presented later, if necessary. The relevance policy ties the explanatory process to the explainer's goals/objectives, stating that information that is more likely to be relevant to the explainer (to achieve his or her goals) should be presented before less relevant information. Hence, one expected effect of the relevance policy is that explanations will be shorter.

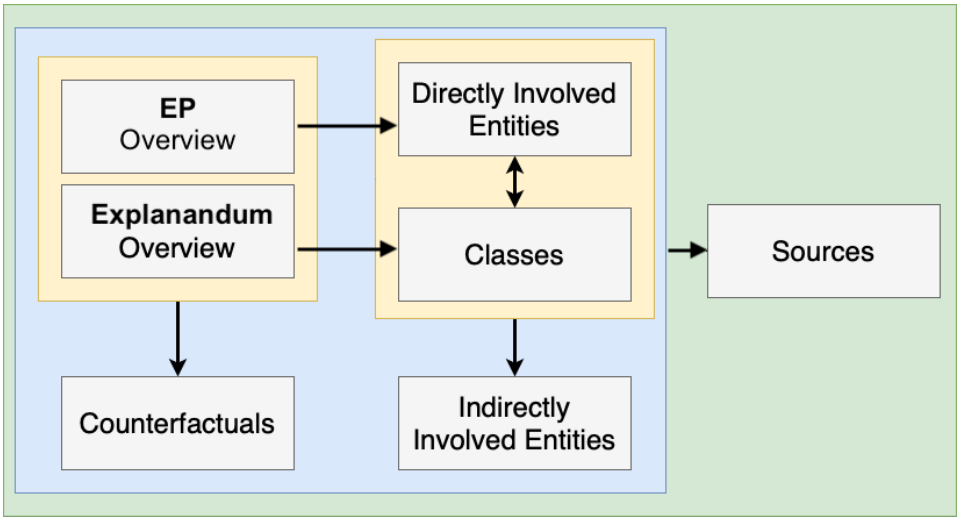
In this scenario, the initial explanans should therefore contain:

- An overview of the underlying explanatory process and explanandum, pointing to meta-data information (e.g., size of the explanandum, language, knowledge representation conventions);

- A brief justification of the automated decision of SPINDle (required in the ex-post phase), perhaps generated through a static explanatory tool such as AIX360 by IBM [96].

In the considered scenario, this justification of the automated decision should indicate that Giulio's request was rejected because of the Italian decree.

Figure 5.3: ***Stylised representation of the structure of the explanatory space of the proof of concept of Section 5.5.*** *In this figure, the arrows show the flow of information, while every rectangle represents a different sub-stage of the space. The flow begins with an overview of the explanatory process (EP for short) and explanandum.*



Considering that the explanatory process is an instantiation of the SAGE-ARS model, for every SAGE property, it is necessary to identify a *set of commands* that the explainee may use during the explanatory process:

- *Sourcing* (commands): used to show the sources, i.e., the Legal-RuleML representation of the law.
- *Adapting*: used to keep track of important information while exploring the explanatory space, building an argumentation framework.
- *Grounding*: used to ask questions, e.g., what would happen if facts would be different; what is the GDPR, and so on.
- *Expanding*: used to add (or remove) pieces of information to the explanations.

Consequently, after defining the initial explanans and the explanatory process, the structure of the explanatory space is also defined. The resulting structure (shown in Figure 5.3) is composed of the following *six main stages*, where each stage of the structure is involved in a different step of the explanans construction:

- *Incipit* (explanatory process and explanandum overview).
- *Core information* (i.e., directly involved entities/classes).
- *Marginal information* (i.e., indirectly involved entities/classes).
- *Counterfactual information*.
- *Source information*.

So, in our case, the explainee (Giulio) wants to get an explanation of the conclusions taken by SPINdle (the decision of the automated decision-making process). To do so, Giulio can use the aforementioned explanatory process, thus reaching the initial explanans, shown in Figure 5.4, which provides a succinct *ex-post* justification of the decision taken by SPINdle. This justification points to further explanations about relevant concepts and information in the explanandum. However, this first-level explanation is not

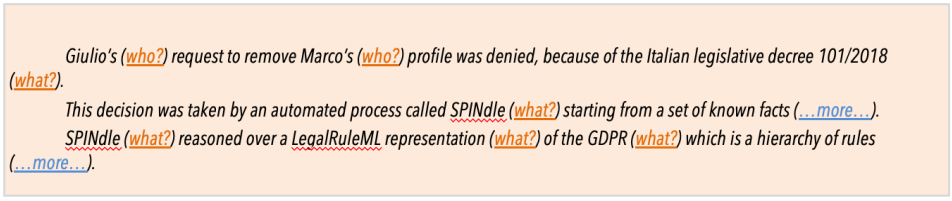


Figure 5.4: ***Incipit stage of the proof of concept YAI of Section 5.5: initial explanans.*** *In this figure, a simple example of initial interactive explanans is shown. Coloured underlined text within round brackets represents different SAGE commands. When clicking on them, the explainee can interactively change the content of the explanation on-demand, exploring the explanatory space.*

enough for Giulio. Thus he clicks on a "what?" button to understand what is the GDPR, thus moving to the *core information* stage (directly involved entities), as shown in Figure 5.5. Now, Giulio wants to get more information about the decision process. Thus he clicks on the "(...hide...)" button and then on a few "(...more...)" buttons to further expand the initial explanans. This time Giulio finds out which sequence of rules was applied by SPINdle to produce the decision. So, now he can see that every rule is linked to a LegalRuleML component and the relevant source of law that justifies it. Giulio can also see rebuttals, as shown in Figure 5.6. If he requested the explanatory process to tell more about the GDPR rebuttal, he

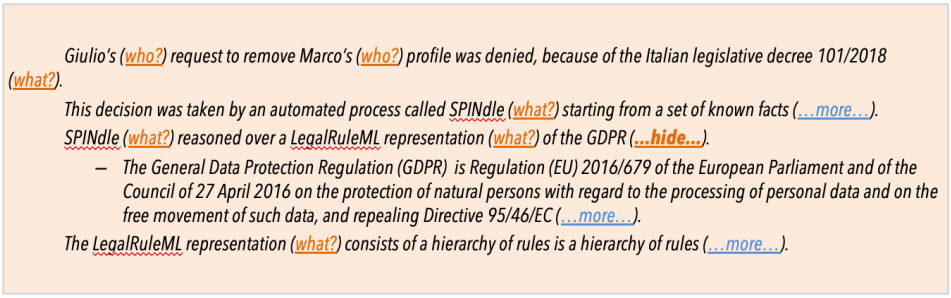


Figure 5.5: ***Core information stage of the proof of concept YAI of Section 5.5: explanations about the GDPR.*** *Grounding commands are shown in orange, while expanding commands are in blue.*

would find out that the "lex specialis derogat generali" was applied, causing the activation of the rule associated with the Italian decree instead of the rule associated to the GDPR. In addition, Giulio also wants to know

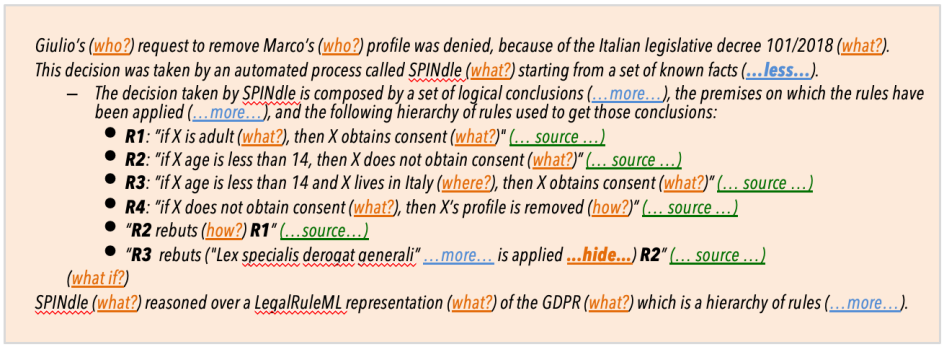


Figure 5.6: ***Proof of concept YAI of Section 5.5: explanations about the decision process.*** *This figure shows the causal chain of the automated decision. Clicking on the "(what if?)" button, it is possible to move to the counterfactuals stage, while clicking on the "(...source...)" buttons, it is possible to see the LegalRuleML sources. Grounding commands are shown in orange, expanding commands are in blue, while sourcing commands are in green.*

more about the set of premises. Thus he clicks on another "(…more…)" button, seeing that the residence in Italy is one of the premises used by SPINdle to take its decision, as shown in Figure 5.7. Then, Giulio clicks on the "what if?" button or on the "(...find alternative...)" button to reach the counterfactual stage and to understand what would happen if Marco's

residence were different. This way, Giulio understands that if his family moves to Luxembourg (where the minimum age for giving consent without parental authorisation is 16) for three months, then a European judge could reassess the case giving Giulio the right to decide for Marco.

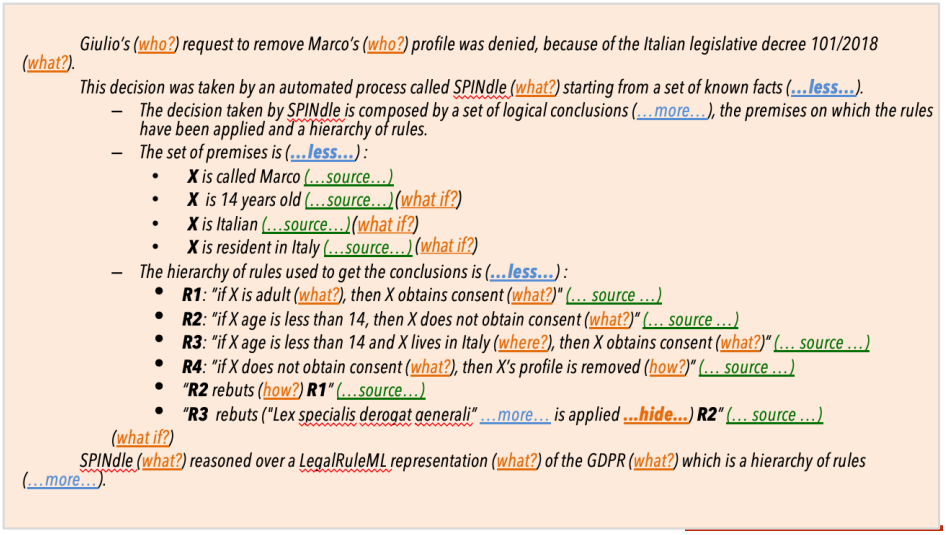


Figure 5.7: ***Proof of concept YAI of Section 5.5: decision process premises.*** *This figure shows the premises as well as the causal chain of the automated decision. Clicking on the "(...find alternative...)" buttons, it is possible to move to the counterfactuals stage, while clicking on the "(...source...)" buttons, it is possible to see the LegalRuleML sources. Grounding commands are shown in orange, expanding commands are in blue, while sourcing commands are in green.*

With an explanatory tool based on our model, the user can explore the explanatory space and build his explanatory narrative through a set of pre-defined actions. The resulting tool is user-centred by design and can be used for finding evidence to make sense of phenomena (sense-making), articulating understandings into an explanatory narrative. We assert that the structure of the explanatory space we have identified is sufficient to produce the descriptive, causal, and justificatory explanations mandated by the GDPR. Specifically:

- *Descriptive explanations* can be derived from the core stage and the marginal information stage;

- *Causal explanations* can be obtained through the counterfactuals stage;
- *Justificatory explanations* can be achieved by examining the incipit stage and the sources stage.

The proof of concept demonstrates the potential of the SAGE-ARS model for developing a GDPR-compliant YAI system, although certain limitations must be acknowledged. Specifically, the proof of concept does not incorporate all the primitive actions associated with SAGE commands, as discussed in Section 5.4, particularly open-ended question-answering. This limitation may affect the depth and detail of explanations provided. While this approach works well for the specific scenario considered, it might not be universally applicable to all XAI or YAI situations. Users may need to search for information relevant to their particular query, as the model's lack of parsimony could create additional steps in finding the desired explanation. However, for the identified real-world scenario, the proof of concept demonstrates the feasibility of using the SAGE-ARS model to develop a YAI that offers explanations compliant with the GDPR.

By adhering to simplicity, abstraction, and relevance heuristics, the YAI enables users like Giulio to navigate complex legal information and comprehend automated decision-making processes. The tool also allows users to interactively explore the explanatory space and access pertinent information tailored to their needs. This approach not only helps users understand the decision process but also fosters trust in the system, ensuring compliance with GDPR requirements and promoting transparency in automated decision-making.

In the subsequent chapters, we will discuss more comprehensive examples of YAI implementations, including those featuring the open-ended question-answering primitive action.

Part II

Explanatory Tools for Humans: Experiments and Empirical Results

Summary

In this part of the thesis, we provide strategies, intelligent interfaces and other software tools to answer the fourth research question posed in the introduction: how to generate explanations for humans in an algorithmic way.

In Chapter 6, we show how to use question-answering algorithms to implement Explanatory Artificial Intelligence software. We then discuss the implementation of YAI4Hu, a YAI tool for humans that relies on intelligent components for open-ended questions and aspect overview. To evaluate YAI4Hu, we design and present the results of two user studies in Chapter 7. The empirical results show that YAI4Hu surpasses all identified baselines, supporting our theory. Next, in Chapter 8, we show how to use some components of YAI4Hu to implement DoXpy, an algorithm capable of quantifying the degree of explainability described in Chapter 4.

Several approaches to automatic question answering exist, but not all of them can scale from small to extensive and heterogeneous explananda or are fast enough to answer any question quickly with standard computing capabilities. Therefore, in Chapter 9, we identify strategies for scalable question-answering on technical languages. These strategies rely on linguistic theories, data mining and knowledge extraction techniques to efficiently retrieve answers without training or fine-tuning procedures. Finally,

The content of Part II is a reworking and extension of the following articles by the same author of this thesis: [194, 195, 188, 189, 199, 190, 200].

in Chapter 10, we show how to construct YAI tools for education to explain excerpts of a legal textbook. In particular, this part of the thesis is intended to show how our technology can work with both technical (i.e., legal) and ordinary languages, processing and explaining a wide range of documents.

CHAPTER 6

How to Use Question-Answering Algorithms to Implement the SAGE-ARS Model

If explaining is about answering questions, we can build an explanatory tool on top of question-answering algorithms. From the definition of explanatory illocution given in Section 4.1.1, it follows that illocutionary question-answering requires a mechanism for pragmatically:

- *Estimating the pertinence* of answers to (archetypal) questions;
- *Identifying the relevant aspects* to be explained through illocution.

The problem with this is that every user may need different information depending on her or his background knowledge, making it very hard to estimate the pertinence of informative content, at least pragmatically speaking.

To solve this problem, we frame *pertinently answering* as the process of giving answers that are likely to be relevant to a given (archetypal) question. The likelihood can be quantitatively estimated on strong-enough statistical evidence collected from large corpora and built in language models.

The point is that this statistical definition of pertinence is compatible with Achinstein's *u-restrictions* (introduced in Section 2.2), and it does not preclude a user-centred explanatory process that is locally non-user-centred but globally user-centred.

It is possible to see the space of all the explanations about an explanandum (or explanatory space) as a sort of manifold space where every point within it is interconnected explainable information that is not user-centred locally (because it is the same for every user), but globally as an element of a sequence of information that can be chosen by users according to their interest drifts while exploring the space. Importantly, this understanding of an explanation as a sequence within the explanatory space is indeed framing explanations as *ellipses* (a concept introduced in Section 2.2), for the explanation being a pragmatic subset of all the possible information about an explanandum.

It is possible to see the explanatory space as a hypergraph of interconnected bits of explanation (see Section 5.3 for more details) and an explanation as a path within the explanatory space that can be generated throughout question-answering. Consequently, the relevant aspects to be explained are framed as clusters of these interconnected bits of explanation. For example, assuming that the explanandum is a set of documents written in a natural language (e.g., English), the relevant aspects to explain might be the different words within the corpus so that an explanatory overview can be associated with each word. So, in order to implement the ARS heuristics and the SAGE actions (see Section 5.4), we may use an algorithm able to:

1. Extract from a corpus of documents a (knowledge) graph representing the different aspects (words) to be explained and the information related to them;

2. Organise the explanandum aspects as a taxonomy[1], thus ordering them by level of *abstraction*;

3. Answer open-ended (English) questions *relevant* to the explainee, possibly exploiting a taxonomy of aspects to drastically reduce the size of the search space from the whole corpus of documents to only those snippets mentioning aspects related to the question;

[1] A taxonomy is a hierarchical tree classification scheme in which things are organised in terms of relationships between subclasses.

4. Build at least one information cluster (or explanatory overview) per aspect, ordering information within the cluster by *relevance* to archetypal questions;

5. Hide redundant information within clusters, favouring shorter, *simpler* and most informative explanatory overviews.

In the following sections, we will present a practical implementation of such an algorithm, called YAI for humans (YAI4Hu for short).

YAI4Hu is a fully automatic explanatory tool built on the model described in Chapter 5. YAI4Hu is capable of enhancing the explanatory power of a collection of (English) texts, representing it as a hypergraph where information can be either explored through *overviewing* or searched via *open-ended questioning*. On the one hand, *open-ended questioning* can be performed by asking questions (in English) through a search box that uses the (knowledge) graph for efficient answer retrieval. On the other hand, explanatory *overviewing* can be performed iteratively from an initial explanation by clicking on (automatically) annotated words that require an explanation.

One notable limitation of the current YAI4Hu version is its inability to generate new information. Its functionality is limited to extracting pre-existing data, meaning it can only provide information that has been pre-recorded and does not have the capacity to rephrase it. This is because the YAI4Hu framework currently lacks generative AI components. Nevertheless, a potential solution to overcome this limitation lies in the future integration of advanced generative AI technology. By incorporating such technology, YAI4Hu could be equipped not only to extract information but also to generate and rephrase it. This advancement would significantly expand the capabilities of the tool.

6.1 Efficient Answer Retrieval

The task of answering questions using an extensive collection of documents about diverse topics or from different domains is called open-domain question-answering [47, 94]. There are at least three main software architectures for open-domain question-answering: the retriever-reader, the retriever-generator and the generator-only architecture. The first two architectures combine information retrieval techniques and neural reading com-

prehension or text generation models. In particular, the latter does not involve classical information retrieval, thus being completely end-to-end.

A famous example of generator-only architecture could be OpenAI's ChatGPT[2], an adaptive and intelligent dialogue system. This type of algorithm usually relies on large deep neural networks that are trained in an unsupervised manner to memorise facts and in a supervised manner to answer questions in a meaningful and coherent way. Even though generator-only architectures are capable of impressive results, they tend to write plausible-sounding but incorrect or nonsensical answers. One of the reasons for this problem is that this type of architecture is fully end-to-end and needs to perform fact-checking.

In contrast, the retriever-generator and retriever-reader architectures circumvent the latter problem by relying on a system capable of retrieving plausible answers from a knowledge base (or graph) of verified contents. The retriever-reader and retriever-generator models usually have an asymptotic time complexity that grows linearly with the number of answers considered for retrieval. That number does not necessarily have to be equal to the number of all the retrievable texts. In other words, the time complexity of the answer retrieval system can be intelligently controlled by making it fit the memory and time constraints of a personal computer, e.g., by filtering out all texts unrelated to a question. In particular, the retriever-generator rewrites and reprocesses the retrieved information, while the retriever-reader limits itself to extracting it (as it is) and reclassifying it properly.

This thesis focuses not on question-answering or chatbot technology but on how to model and design user-centred AI regardless of the algorithm pipeline adopted. In other words, at this research stage, we are not interested in end-to-end opaque explanatory systems but rather in building an Explanatory AI that reflects the identified theoretical insights and allows us to verify them empirically. Therefore, as a question-answering paradigm for YAI4Hu, we considered a retriever-reader, shown in Figure 6.1, capable of exploiting the structure of a knowledge graph to filter answers in order to perform an efficient (i.e., on average, sub-linear) search across large amounts of text. Specifically, a retriever-reader architecture was chosen because it is easier to implement and execute with standard computing capabilities (e.g., a desktop or laptop computer). Moreover, it does not require a generator or the fine-tuning of large and opaque deep

[2] `https://openai.com/blog/chatgpt/`

learning models with expensive annotated datasets. Indeed, intuitively, text generation is a more complex task than reading comprehension.

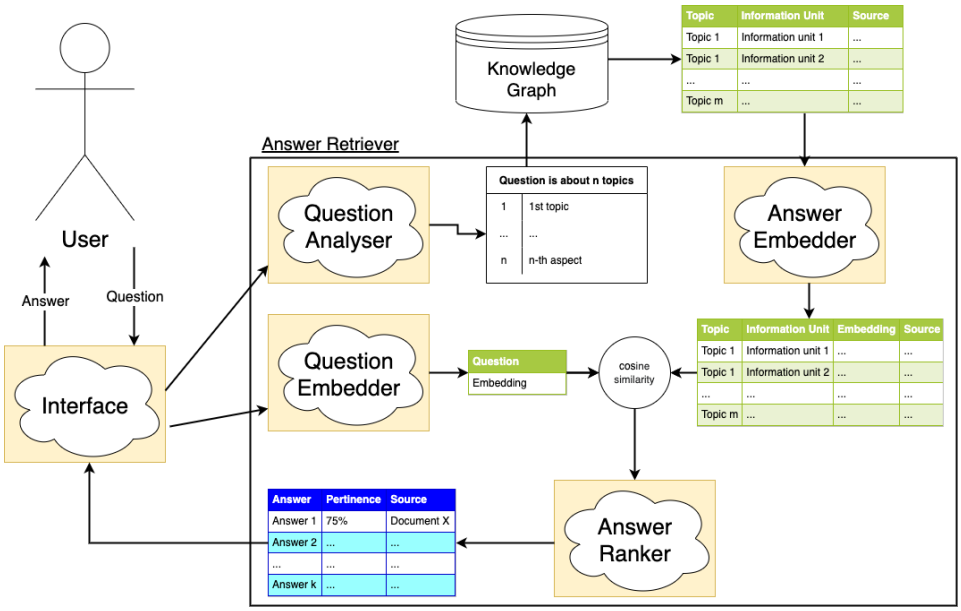


Figure 6.1: ***Flow diagram of the answer retriever used by YAI4Hu.*** *This figure summarises the answer retrieval algorithm used by YAI4Hu. A question q about one or more topics C (i.e., explanandum aspects) is given as input to the retriever, which analyses it to find the set of information units about C in the knowledge graph. Then, both the identified information units and q are embedded, and their cosine similarity (θ) is used to rank information units according to their pertinence to the question, selecting them as answers.*

Open-domain answer retrieval systems (also called dense passage retrieval or question-answer retrieval [47]) based on reading comprehension encode all the identified possible answers (e.g., parts of sentences, paragraphs) into a numerical representation (i.e., a vector of real numbers) with a general-purpose neural model. Then they use the encoding for fast similarity-based retrieval. Amongst the most important answer retrieval models, we distinguish between those that use the answer context for the generation of embeddings[3] [225, 106, 172] and those that do not [48].

[3]Intuitively, using the answer context should help the embedder to contextualise and disambiguate better, producing more high-quality embeddings.

The YAI4Hu retrieval-reader mechanism, as depicted in Figure 6.1, employs a method for converting questions and answers into dense numerical representations or embeddings. This approach uses the *cosine similarity* between a question's embedding and an answer's embedding to gauge the relevance of the answer to the question.

More specifically, let T be the set of topics mentioned by a question q and u be an *information unit* (i.e., a grammatical clause, part of a sentence, a sentence) and z the context paragraph from which u was taken. YAI4Hu performs answer retrieval by retrieving all the information units u about T, selecting those likely to be an answer to q. The probability that u pertinently answers q can be estimated as the numerical similarity between the embedding of $u + z$ (i.e., u concatenated with z) and the embedding of q. So that if $u + z$ is similar enough to q, then z is said to be an answer to q for the information unit u. Therefore, in practice, the algorithm can retrieve any arbitrary number of answers, given that enough information units are available.

Figure 7.4 provides an illustrative example of questions and answers retrieved from a specific source, serving as a crucial visual aid in comprehending the retrieval process. By juxtaposing the posed questions with their corresponding responses, the figure allows an easy comparison of inputs and outputs. It also gives insights into the system's capabilities, the complexity of the answers it can generate, and any potential limitations.

The process of obtaining embeddings for $u + z$ and q involves the use of deep language models that specialise in answer retrieval. These models are pre-trained on standard English, enabling them to associate similar vectorial representations with a question and its appropriate answers. Two examples of such pre-trained deep language models are the Multilingual Universal Sentence Encoder [225] and a variant of MiniLM [219], as published by SBERT [165]. YAI4Hu specifically employs the Multilingual Universal Sentence Encoder, developed by Yang et al. [225], and trained on the Stanford Natural Language Inference corpus [28].

One problem of models trained on ordinary English is that they tend to lose effectiveness whenever applied to more technical (e.g., legal, scientific) languages, as those considered in practically all our case studies. For this reason, Chapter 9 discusses some techniques to circumvent this issue without expensive training procedures or datasets.

6.2 Automated Graph Extraction

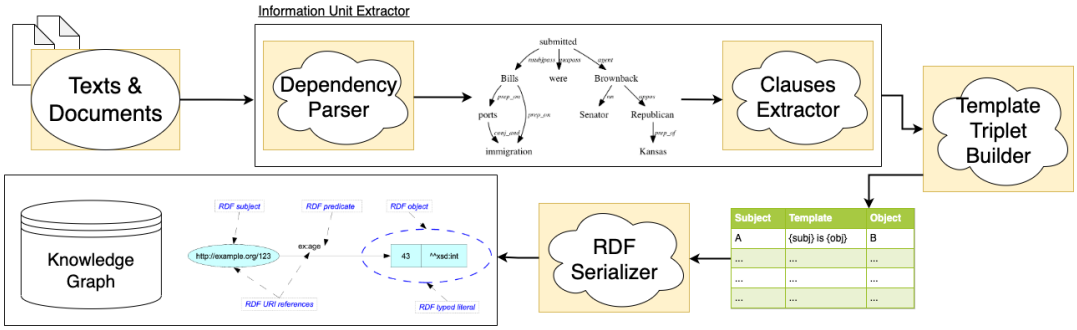


Figure 6.2: ***Flow diagram of the automated graph extractor used by YAI4Hu.*** *This figure summarises the algorithm used by YAI4Hu to extract the knowledge graph for answer retrieval, as described in Section 6.2.*

Knowledge graph extraction is the extraction of concepts and their relations, from natural language text, in the form of a graph where concepts are nodes and relations are edges. When these relations are complex clusters of information connecting more than two concepts (i.e., whole clauses or sentences), the knowledge graph is called a *hypergraph*. Thus, we are looking for a way to extract knowledge hypergraphs whose relations would preserve the original natural language. So that we can make the graph easily interoperate with the answer retrieval algorithms and existing deep language models described in Section 6.1.

As shown in Figure 6.2, such hypergraphs can be extracted by detecting, through a dependency parser[4], all the possible clauses in every sentence of the corpus of documents considered as explanandum. By analysing the grammatical dependencies extracted by a dependency parser, it is possible to identify subjects and objects as explanandum aspects (or nodes) of the hypergraph, and the clauses or sentences[5] that contain them as hyperedges. Specifically, *dependency parsing* involves exploring the dependencies between words in a sentence to understand their grammatical structure by

[4]For example: `https://spacy.io/api/dependencyparser`

[5]Complex sentences contain multiple clauses including at least one independent clause (meaning, a clause that can stand alone as a simple sentence) coordinated either with at least one dependent clause (also called an embedded clause) or with one or more independent clauses.

breaking sentences into multiple components linked together by grammatical dependencies (e.g., subject, object, adverb, verb) to form clauses.

In our case, for simplicity and efficiency, we decided to represent these clauses as special triplets, called *template-triplets*, containing a: subject, template, and object. Specifically, templates are composed by the ordered sequence of tokens connecting a subject to an object (or nominal modifier[6]) in a clause. In particular, the subjects and the objects are represented in these templates with the placeholders *"{subj}"* and *"{obj}"*. An example (taken from the proof of concept of Section 5.5) of template-triple is:

- Subject: *"Giulio's request"*
- Template: *"{subj} to remove Marco's profile has been denied because of {obj}"*
- Object: *"the Italian legislative decree 101/2018"*

Alternatively, these template-triplets can be seen as a function, where the template is the body and the object and the subject are the parameters. So that obtaining a natural language representation of them (i.e., an information unit to be used for answer retrieval) is straightforward by design, replacing the instances of the parameters in the body.

Eventually, collecting all these template-triplets makes it possible to extract a binary decomposition of a knowledge hypergraph from any collection of textual documents. Each template is a binary edge (between a subject and an object) standing for a hyperedge which connects the subject and the object as well as all the other concepts and entities mentioned in the template.

In order to increase the interoperability of these knowledge graphs with external resources, it is also possible to serialise them as RDF graphs. RDF is a standard model for binary graph representations and data interchange on the Web [6]. Importantly, RDF has features that facilitate data merging even if the underlying schemas differ. This property of *compositionality* of RDF graphs (i.e., two RDF graphs can be combined without complicated merging procedures) gives the possibility to manually correct any error produced during the graph extraction and easily integrate it with additional knowledge (graphs) in order to manually fine-tune the behaviour of an answer retriever.

[6] A nominal modifier is a noun (or noun phrase) functioning as a non-core (oblique) argument or adjunct.

In order to format a graph of template-triplets as an RDF graph, it is sufficient to:

- Assign a *Unique Resource Identifier* (URI, for short) to every node (i.e., subject and object) and edge (i.e., template) of the graph by lemmatising the associated text. Each URI points to an RDFS label corresponding to the associated text;
- Add special triplets to keep track of the *sources* from which the template-triplets were extracted so that for each node and edge is possible to go back to the source document or paragraph;
- Add a *sub-class* or *is-an-instance-of* relation between composite concepts (syntagmas) and the simplest concepts (if any) composing the syntagm. For example, "contractual obligation" is a sub-class of "obligation", while "Giulio's request" is an instance of "request".

6.3 Overview Generation via Answer Retrieval

As discussed in Section 5.4, an overview is an explanation consisting of a sequence of relevant answers to (possibly interesting) archetypal questions about an explanandum aspect. In particular, overviewing is the primitive action identified for *articulating understandings* and allowing the user to explore an explanatory sub-space iteratively. An example of an overview generated by ChatGPT, answering several archetypal questions about Christopher Columbus[7], is:

> *"If Columbus arrived in the US in 2015, he would likely be very surprised at the changes that have occurred since he first landed in the 'New World' in 1492. For one, he would probably be shocked to discover that the land he 'discovered' was already inhabited by Native Americans and that now the United States is a multicultural nation with people from all over the world. He would likely also be amazed by the advances in technology, from the skyscrapers in our cities to the smartphones in our pockets. Lastly, he might be surprised that many people*

[7] Christopher Columbus was an Italian explorer and navigator who completed four voyages across the Atlantic Ocean, opening the way for the widespread European exploration and colonisation of the Americas.

> *do not view him as a hero anymore; some argue that he was a brutal conqueror who enslaved and killed native people. Overall, it would be a very different experience for Columbus than the one he had over 500 years ago."*[8]

According to Hypothesis 1 (cf. Section 3.2), it is possible to generate an explanatory overview through the answer retrieval algorithm described in Section 6.1 by clustering and ordering information according to its pertinence to a set of (predefined) archetypal questions, such as those listed in Section 3.3. Answering different archetypal questions implies providing different explanatory nuances that can shed light on causes and justifications (if the considered archetype is `why`) as well as finality (`what for`), time (`when`), space (`where`), properties (`what`), and so on.

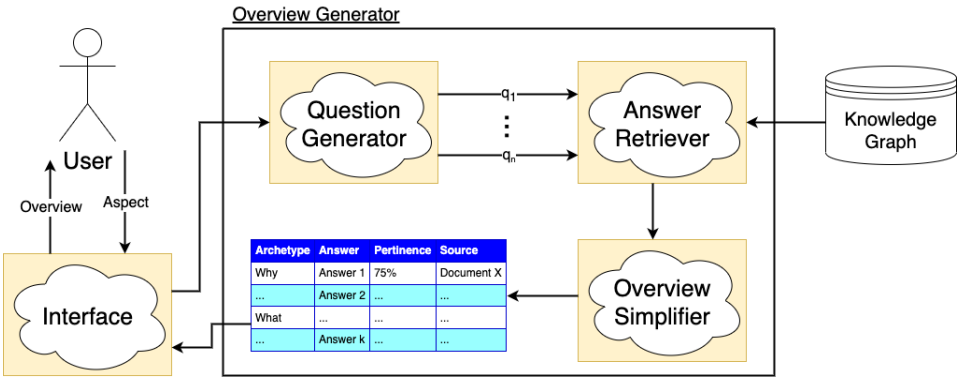


Figure 6.3: ***Flow diagram of the overview generator used by YAI4Hu.*** *As shown in this figure, once the user selects an explanandum aspect to overview, the question generator identifies a meaningful set of (archetypal) questions about that, and it uses them for organising the overview through the answer retriever described in Section 6.1. Resulting overviews are simplified, removing redundant information as explained at the end of Section 6.3.*

In practice, an algorithm for overview generation can be obtained by simply piping a question generator with an answer retriever and an algorithm (called overview simplifier) for adequately cleaning and formatting the output of the answer retriever, as shown in Figure 6.3. The question generator takes as input an explanandum aspect (e.g., "*constitution*"). It

[8]This snippet of text has been taken from `https://openai.com/blog/chatgpt/`

produces as output a set of archetypal questions about that aspect (e.g., "*What is the constitution?*", "*What is the purpose of the constitution?*"). In particular, YAI4Hu uses a naive but effective question generator that employs a predefined set of templates of archetypal questions (e.g., "*What is {x}?*", "*Why {x}?*"), replacing the template placeholder with a label of the explanandum aspect (e.g., the "*{x}*" in "*What is {x}?*" is replaced with "*constitution*").

Let Q be a set of archetypal questions, $q \in Q$ an archetypal question, and c an explanandum aspect, YAI4Hu generates an explanatory overview by: *extracting*, from the knowledge graph produced by the algorithm of Section 6.2, all the information units related to c, including those of the sub-classes or instances of c; and *selecting*, through the answer retriever described in Section 6.1, the information units that are more likely to be an answer to q, for each $q \in Q$.

Considering that an answer might be associated with more than one archetypal question, the overview simplifier is responsible for filtering out redundant answers, according to the *simplicity* heuristic. Specifically, if the answer retriever gives the same answer to more than one question, the overview simplifier assigns it to the most pertinent question. Moreover, the overview simplifier is also responsible for sorting the questions by decreasing pertinence (i.e., the questions whose first answer has the highest pertinence score are ranked first) and guaranteeing that each question does not have more than a answers (a is called answer horizon).

To further guarantee the abstraction policy, the overview simplifier could also integrate the overview with taxonomical information (if any) about the explanandum aspect, e.g., super-classes, sub-classes and instances.

For a practical demonstration of these principles in action, refer to Figure 7.5 in Section 7.2 where an example of such an overview is provided.

6.4 Smart Annotation: Selection of Which Aspects to Explain

An explainee can visualise overviews by selecting an aspect to overview (e.g., by clicking on annotated words on the screen). However, not all words are aspects for which it makes sense to produce an explanatory overview. That is because, in practice, only a tiny fraction of the words in a text are helpful to explain. Indeed, the meaning of many words belongs to common sense (e.g., the words: "*January*", "*and*", "*first*", "*figure*") and therefore

should not be explained. Otherwise, the explainee might be overwhelmed by largely redundant and pointless information that would only hinder the usability of the YAI. A good YAI should be able to avoid redundant explanations and favour shorter, *simpler* and most informative explanatory contents, thus making use of *ellipses* (cf. Section 2.2).

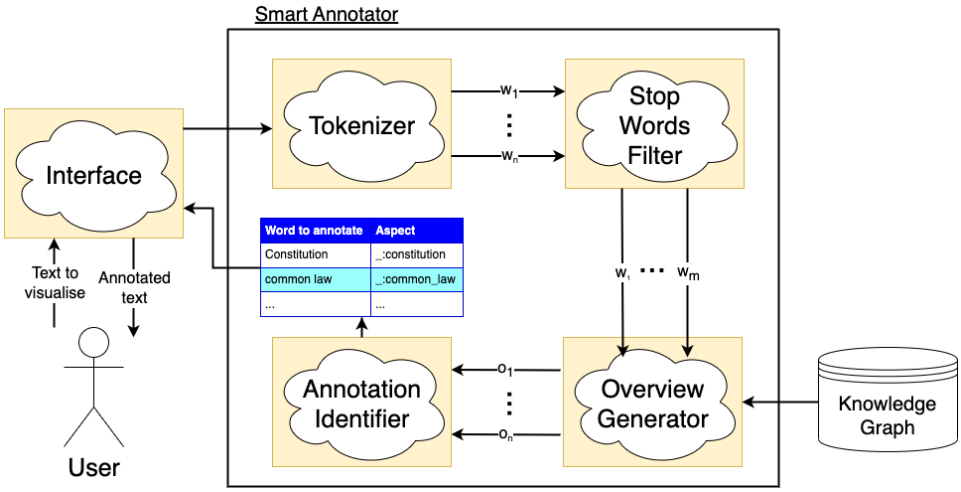


Figure 6.4: ***Flow diagram of the smart annotator used by YAI4Hu.*** *As shown in this figure, every time the user visualises some explanation provided by YAI4Hu, the textual content of it is sent to the smart annotator. The smart annotator tokenises the input text, looking for words to annotate so the user can click on them and overview them. Examples of automatically annotated words (i.e., the underlined words) are shown in Figure 7.5.*

Therefore, to intelligently avoid producing unnecessary explanations, YAI4Hu enforces that users can only overview the most explainable words. These words are shown on the screen with a special distinguishable mark (i.e., an annotation; an underline) automatically generated by YAI4Hu. Specifically, the annotation mechanism of YAI4Hu (summarised in Figure 6.4) annotates only those concepts and words that are the most explained by the content of the knowledge graph used for answer retrieval. More specifically, to understand whether a word should be annotated, the algorithm executes the following instructions.

1. It identifies words using a tokeniser.

2. It checks whether the word is a *stop word* (i.e., a commonly used word such as "and" or "or") and if that is the case, then the word is not annotated. Specifically, stop words are associated with general knowledge of the world, and they can be heuristically identified by analysing word frequency in the Brown corpus [181] or similar corpora.

3. If the word is not a stop word, the algorithm generates its overview, and it passes it to the annotation identifier.

4. The annotation identifier aggregates the pertinence scores of the answers composing the overview, thus computing the *cumulative pertinence score*. If the cumulative pertinence score of a word overview is greater than a given threshold o, the word is selected for annotation and associated with a node of the knowledge graph (i.e., an explanandum aspect).

This annotation mechanism is intended to significantly remove noisy annotations and distractors so that the reader can focus only on the most central and well-explained concepts. Moreover, the cumulative pertinence score used to understand whether a word should be annotated can also be used to understand the most explained topics in the corpus of documents. The cumulative pertinence score is a variation of the average DoX described in Chapter 4.

CHAPTER 7

Experimental Validation of the SAGE-ARS Model: YAI vs One-Size-Fits-All Explainers

Our proposed model of YAI is defined around the idea that explaining is somewhat akin to exploring a possibly unbounded hypergraph of questions and answers called explanatory space. This explanatory space, to be efficiently explored through an explanatory process, is then broken down into a kind of hypertree decomposition to allow the explainee to navigate through the vast underlying space and find the answers he or she is looking for.

In particular, Hypothesis 3 (cf. Section 5.4) states that an explanatory process implementing the ARS heuristics and the SAGE commands produces better explanations than any one-size-fits-all explainer, generating an easy-to-navigate decomposition of the explanatory space. In other words, Hypothesis 3 is equivalent to saying that not all the decompositions of an explanatory space are equally useful to a human subject (if no assumption is made about the background knowledge of the explainee), and that the SAGE-ARS model can produce a decomposition that is user-centred and

useful (e.g., for data subjects). Moreover, Hypothesis 1 (cf. Section 3.2) further specifies that the usability (as per ISO 9241-210) of these decompositions is directly affected by the illocutionary force and goal-orientedness of the explanatory process.

In order to test these two hypotheses, we gathered some empirical results, which are presented in this chapter. Specifically, the hypotheses were tested through between-subjects user studies involving hundreds of participants and comparing the user-centrality (measured in terms of usability) of baseline, one-size-fits-all explanatory tools with that of tools adhering to the SAGE-ARS model (i.e., YAI4Hu). As case studies for the evaluation, we considered two separate scenarios in which the automated decisions of AI-based systems for credit approval and prediction of heart diseases are explained to laypersons for compliance with the law (cf. Chapter 1) or to help them achieve their objectives.

7.1 The Explananda: Two XAI-Based Systems for Finance and Healthcare

The two AI-based systems we considered as case studies concern healthcare and finance. They are, respectively: a *heart disease predictor* based on XGBoost [49] and TreeSHAP [126]; a *credit approval system* based on a simple artificial neural network and CEM [62].

7.1.1 Finance: the Credit Approval System

IBM designed the credit approval system under consideration to present AIX360[1]. It uses an artificial neural network to predict a customer's credit risk (and thus decides whether or not to approve a loan) together with an XAI algorithm (called CEM [62]) to provide post-hoc static explanations of the neural network's predictions. These explanations aim at helping customers understand whether they have been treated fairly, providing insights into ways to improve their qualifications so as the likelihood of future acceptance can be increased.

A typical use case of this system is the following one. A customer (e.g., John) applies for a loan from the bank. The bank collects sufficient information about the customer. It transmits it to the artificial neural network,

[1] `https://aix360.mybluemix.net/explanation_cust`

which uses it to estimate the probability that the customer will repay the loan. If the customer's credit risk is low, the loan application is approved, but if the credit risk is too high, the system uses CEM to explain why.

The artificial neural network behind this credit approval system is trained on the *FICO HELOC dataset*[2], containing anonymized information about loan applications made by real homeowners, to answer the following question: "What is the decision on the loan request of applicant X?".

The main goal of the users of this credit approval system is to understand the causes behind a loan rejection and what to do to get a loan accepted. It is because of the specific characteristics of this system and the *right to contest an automated decision* set by the GDPR. This is why CEM is deployed to answer the following questions:

- What are the factors to consider to change the result of the application of applicant X?
- How should factor F be modified in order to change the result of the application of applicant X?
- What is the relative importance of factor F in changing the result of the application of applicant X?

Nonetheless, many other relevant questions might be answered before the user is satisfied and reaches his/her objective. Generally speaking, all these questions can be shaped by *contextually implicit instructions* (cf. Section 2.2) set by specific legal or functional requirements, such as those identified by Bibal et al. [22]. These questions may be: "How to perform those minimal actions?", "Why are these actions so important?", and so on.

Interpreting the internal parameters and complex calculations of an AI model such as this credit approval system is not easy. For example, a layperson trying to obtain a loan might undoubtedly be interested to know that her/his application was rejected (by the AI) mainly due to a high number of credit inquiries on his/her accounts (as CEM can tell). However, this information alone might not be sufficient to achieve her/his goals. These objectives may be beyond the reach of the AI, such as understanding: how to effectively reduce the number of inquiries in order to obtain the loan, what type of credit inquiries may affect his status, what is the difference between a hard and a soft inquiry.

[2] `https://fico.force.com/FICOCommunity/s/explainable-machine-learning-challenge?tabset-3158a=a4c37`

To summarise, the output of the credit approval system is composed by:

- *Context*: a titled heading section kindly introducing the user to the system.
- *AI output*: the decision taken by the artificial neural network for the loan application (i.e., "denied" or "accepted").
- *XAI output*: a section showing the output of the CEM. This output consists of a minimally ordered list of factors deemed to be the most important to change for the outcome of the artificial neural network to be different.

A screenshot of a web application implementing this credit approval system is shown in Figure 7.1.

7.1.2 Healthcare: the Heart Disease Predictor

The explanandum of the heart disease predictor is about health, and a first-level responder of a help-desk for heart disease prevention uses the system. More specifically, the first-level responder is responsible for handling the requests for assistance of a patient, forwarding them to the correct physician in the eventuality of a reasonable risk of heart disease.

First-level responders get basic questions from callers; they are not doctors but have to decide on the fly whether the caller should speak to a real doctor. So, they quickly use the heart disease predictor to determine what to answer the callers and the following actions to suggest. In other words, this system is used *directly* by the responder and *indirectly* by the caller through the responder. These two types of users have different but overlapping goals and objectives. It is reasonable to assume that the responders' goal is to answer the questions of a caller in the most efficient and effective way.

The considered heart disease predictor uses an AI algorithm called XGBoost [49] to predict the likelihood of a patient having a heart disease given its demographics (gender and age), health (e.g., diastolic blood pressure, maximum heart rate, serum cholesterol, presence of chest-pain) and the electrocardiographic (ECG) results. This likelihood is classified into three different risk areas: low (probability p of heart disease below 0.25), medium ($0.25 < p < 0.75$) or high. Therefore, XGBoost is used to answer the following questions:

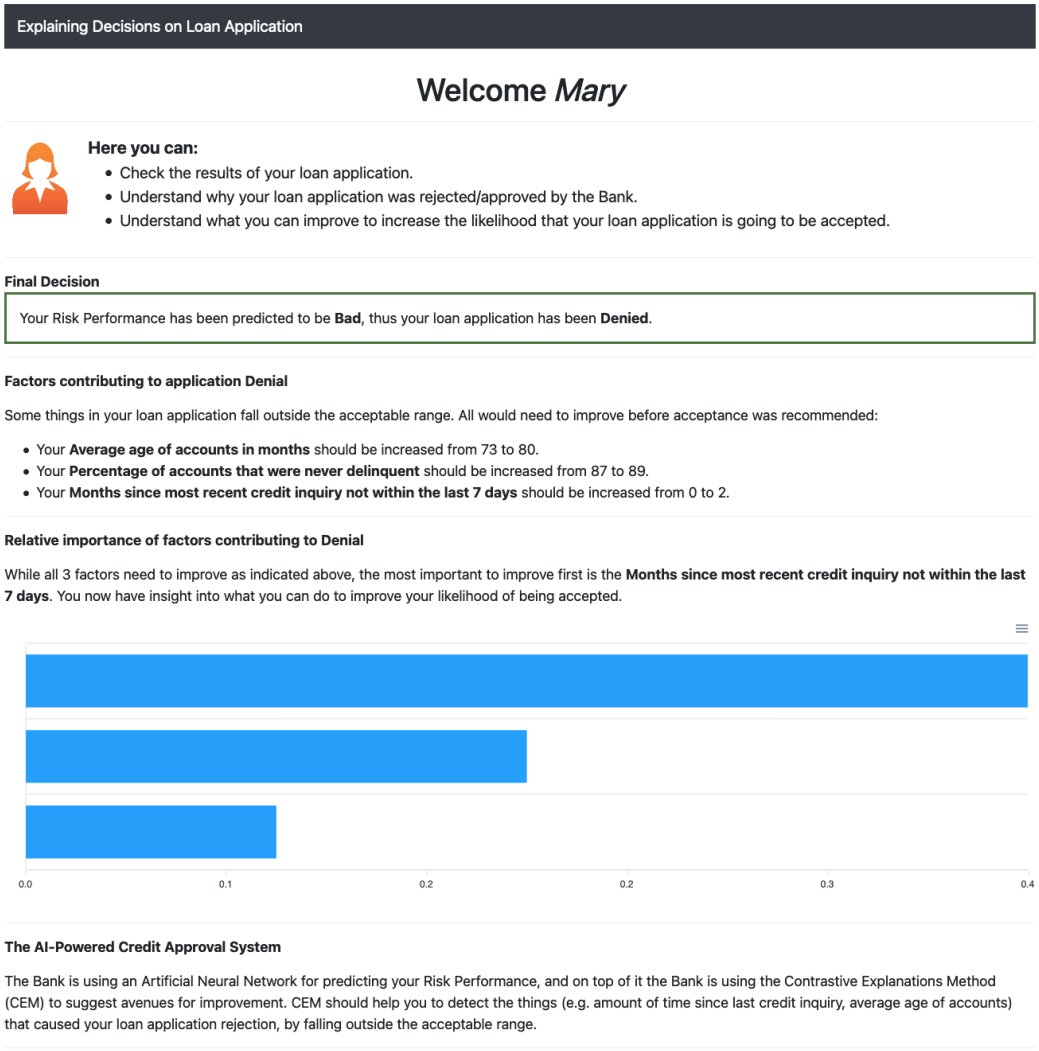


Figure 7.1: ***Screenshot of the credit approval system.***

- How likely is it that patient X has heart disease?
- What is the risk of heart disease for patient X?
- What is the recommended action for patient X to treat or prevent heart disease?

The dataset used to train XGBoost is the *UCI Heart Disease Data* [59, 5].

On top of XGBoost, the heart disease predictor uses TreeSHAP [126], a famous XAI algorithm specialised in tree ensemble models (e.g., XGBoost) for post-hoc explanations. In particular, TreeSHAP is used to understand the contribution of each input feature to the output of XGBoost. Therefore, TreeSHAP is used to answer the following questions:

- What would happen if patient X had factor Y (e.g., chest pain) equal to A instead of B?
- What are the most important factors contributing to the predicted likelihood of heart disease for patient X?
- How factor Y contributes to the predicted likelihood of heart disease for patient X?

However, many other important questions should be answered. These include "What is the easiest thing the patient could do to change his heart disease risk from medium to low?", "How could the patient avoid raising one of the factors, preventing his heart disease risk from raise?".

Finally, to summarise, the output of the heart disease predictor is composed by:

- *Context*: a titled heading section kindly introducing the responder (the user) to the system.
- *AI inputs*: a panel for entering the patient's biological parameters.
- *AI outputs*: a section displaying the likelihood of heart disease estimated by XGBoost and a few generic suggestions about the subsequent actions to take.
- *XAI outputs*: a section showing each biological parameter's contribution (positive or negative) to the likelihood of heart disease generated by TreeSHAP.

A screenshot of a web application implementing this heart disease predictor is presented in Figure 7.2.

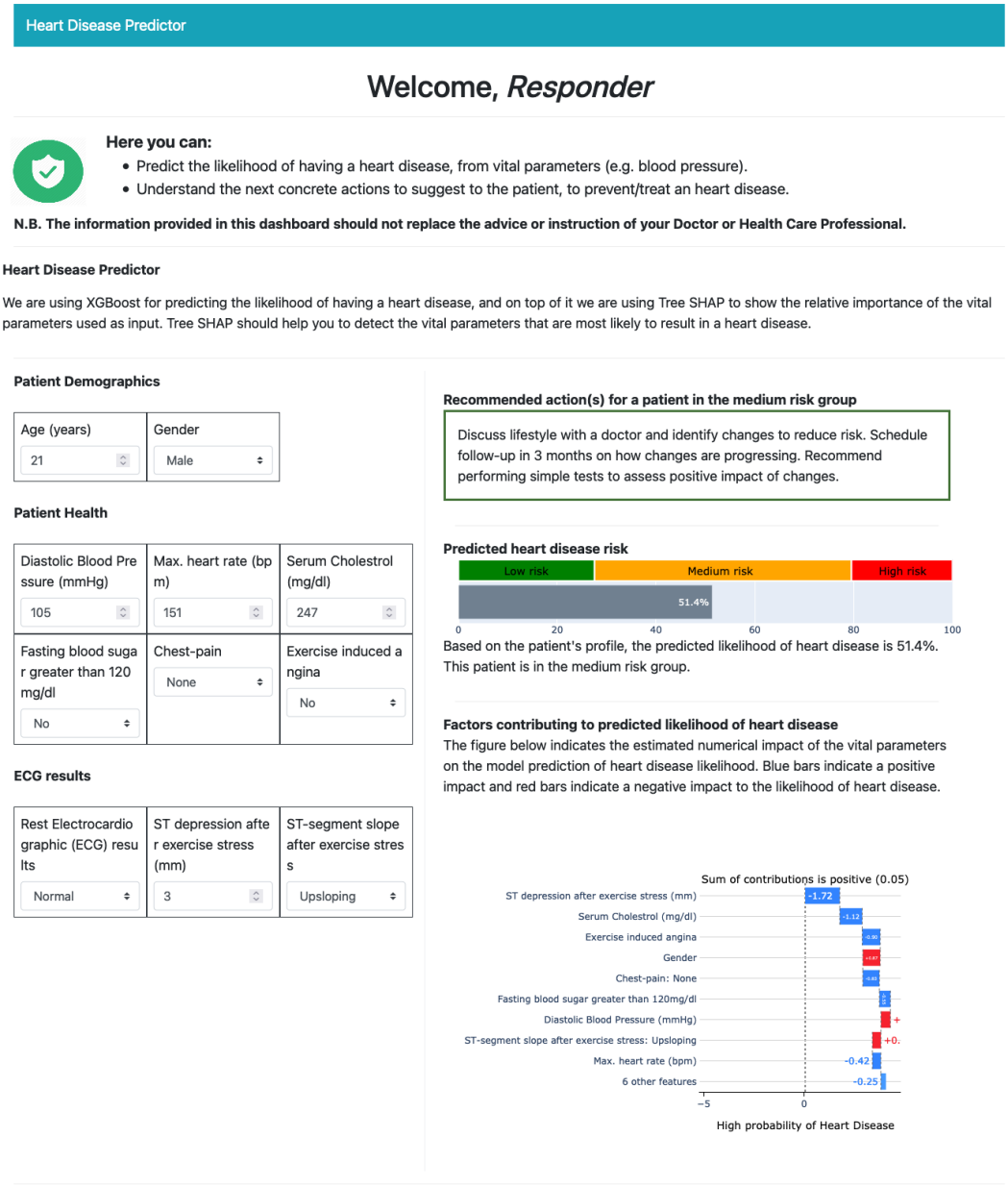


Figure 7.2: *Screenshot of the heart disease predictor.*

7.2 The Explanatory Tools: YAI and One-Size-Fits-All Explainers

During the user studies, the two explananda described in Section 7.1 are explained to human subjects with different tools to find out which explainer is the most usable and thus user-centred. The **explanatory tools** considered for the experiment are implemented as web applications and are the following ones.

A **normal XAI-based explainer**: a one-size-fits-all explanatory mechanism providing the bare output of a XAI as a fixed explanation for all users, together with the output of the wrapped AI, a few extra details to ensure the readability of the results and a minimum of context. Without further changes, the credit approval system and the heart disease predictor are an example of normal *XAI-based explainers*.

A **2nd-level explanatory closure**, or **two-level explainer**, or **overwhelming static explainer**: another static, one-size-fits-all explanatory tool that does not attempt to answer any direct questions. However, unlike normal XAI-based explainers, this type of explainer also uses external documentation to explain the AI-based system (e.g., its features, how to modify them, their meaning, and any background information), dumping large portions of text on the user. Hence, an overwhelming static explainer has more *illocutionary force* than a normal XAI-based explainer, being able to answer a broader set of (implicit) questions but little or no *goal-orientedness*.

A **how-why narrator**: an interactive version of the normal XAI-based explainer designed to provide (on-demand) causal and expository explanations, answering exclusively to `how` and `why` (archetypal) questions through overviewing, and not allowing users to perform open-ended question-answering. The how-why narrator uses and re-elaborates the same external documentation used by the overwhelming static explainer. Therefore, this explainer has less *illocutionary force* (because it can answer only `how` and `why` questions) and more *goal-orientedness* than the two-level explainer (because it allows users to request overviews about topics of their choice).

YAI4Hu (the system described in Chapter 6): an example of YAI approximating a *nth-level explanatory closure* and implementing the

SAGE-ARS model, by answering a wide range of archetypal questions (not just `how` and `why`) through overviewing, also allowing open-ended question-answering. For this purpose, YAI4Hu uses and re-elaborates the same external documentation used by the how-why narrator and the two-level explainer. Thanks to open-ended question-answering, YAI4Hu has more *goal-orientedness* than the how-why narrator. Moreover, thanks to the inclusion of more archetypal questions in its overviews, it also has an *illocutionary force* greater than the previous explanatory tools.

The use (by the two-level explainer, how-why narrator and YAI4Hu) of an extensive collection of external documentation to better explain the two explananda is because the succinct amount of information provided by the AI and XAI algorithms used by the systems is unlikely to be sufficient to cover all the explanatory needs of their users. This is better explained in Sections 5.1 and 7.1. Precisely, the set of external resources carefully selected to cover the topics of the heart disease predictor consists of 103 webpages, 75 of which come from the website of the *U.S. Centers for Disease Control and Prevention*[3], while the remaining from the *American Heart Association*[4], *Wikipedia*, *MedlinePlus*[5], *MedicalNewsToday*[6] and other minor sources. Instead, the external resources used for the credit approval system consist of 58 webpages, 50 of which come from the website of *MyFICO*[7], while the remaining come from *Forbes*[8], *Wikipedia*, *AIX360*[9], and *BankRate*[10]. We took more information (almost double) for the heart disease predictor because, intuitively, it is a more complex explanandum than the credit approval system, requiring much more questions to be covered with different levels of detail.

In the two-level explainer, these external resources are attached to the explanation of the XAI and used as they are (i.e., without any user-driven reorganisation), while the how-why narrator and YAI4Hu intelligently re-elaborate them as overviews or outputs of a question-answering process. Indeed, the two-level explainer is a one-size-fits-all explanatory tool con-

[3] `https://www.cdc.gov`
[4] `https://www.heart.org`
[5] `https://medlineplus.gov`
[6] `https://www.medicalnewstoday.com`
[7] `https://www.myfico.com`
[8] `https://www.forbes.com`
[9] `http://aix360.mybluemix.net`
[10] `https://www.bankrate.com`

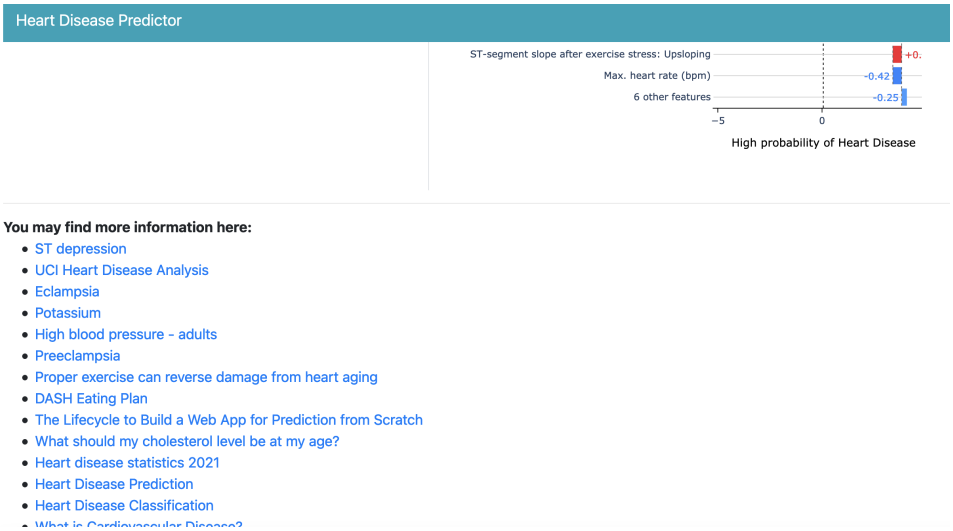


Figure 7.3: ***A 2nd-level explanatory closure for the heart disease predictor.*** *A screenshot showing the connection between the first and the second levels of information used by the two-level explainer for the heart disease predictor.*

sisting of two levels of information, as suggested by the name. The first level is the initial explanans, providing the same output of a normal XAI-based explainer, plus a list of hyperlinks to reach the second level. Instead, the second level consists of a complete and verbose (i.e., more than 50 pages per system, if printed) set of autonomous static explanatory resources for the user to understand the explanandum further. Specifically, the connection between these second and first levels is simply the aforementioned list of hyperlinks, as shown in Figure 7.3.

Differently, the how-why narrator and YAI4Hu augment the normal XAI-based explainers through open-ended question-answering or overviewing. Open-ended question-answering is for users to explicit their own goals and is supposed to be used by those knowing what to ask and how. In other terms, open-ended question-answering is clearly intended as a mechanism for *localisation* of information, and this is possible by entering questions in a simple text input (at the top of the system's landing page; see Figure 7.4), linked to a Python server that exposes the necessary API to interact with the pipeline described in Section 6.1.

In contrast, overviewing is a mechanism for *exploring* information and articulating understandings. Ideally, through overviewing, a user can navi-

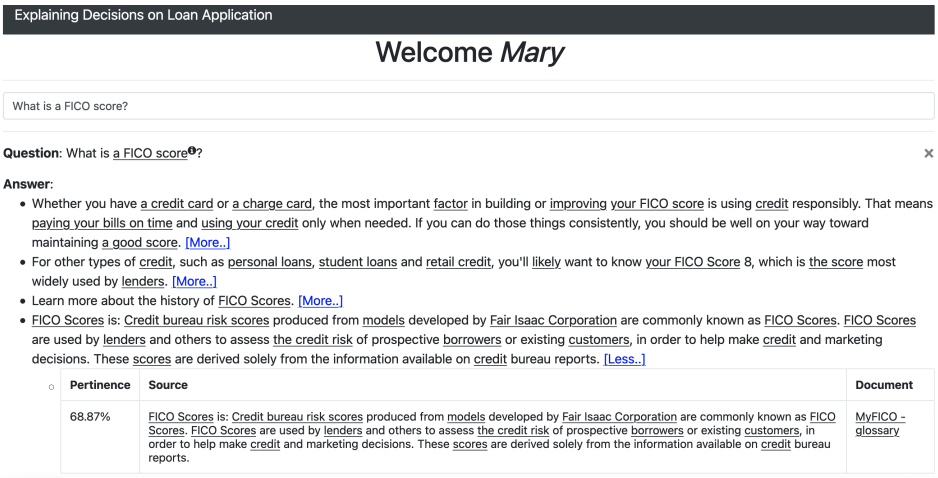


Figure 7.4: ***Answers generated by YAI4Hu for the credit approval system.*** *A screenshot illustrating the process of open-ended question answering within YAI4Hu.*

gate the whole explanatory sub-space reaching explanations for every identified aspect of the explanandum. Therefore, each sentence presented to the user by the web application is automatically annotated through a JavaScript module (described in Section 6.4) that makes the text interactive so that the user can choose which aspect to overview by clicking on the annotated words.

After clicking on an annotation, a modal opens (see Figure 7.5), showing a navigation bar of tabs containing explanatory overviews of the clicked annotated words. The information shown in an overview by the how-why narrator and YAI4Hu is obtained by the systems interrogating a Python server exposing an API to interact with the pipeline described in Section 6.3, and it consists of: *i)* a short description of the explanandum aspect; *ii)* the list of taxonomically connected aspects (i.e., instances, types, subclasses or super-classes); *iii)* a list of archetypal questions and their respective answers ordered by estimated pertinence.

All information shown within the modal is also annotated. This means (for example) that by clicking on the super-class of an aspect, the user can open a new overview (in a new tab) displaying relevant information about it, as shown in Figure 7.5.

As set Q of archetypal questions, YAI4Hu uses pre-defined templates from which questions can be generated by replacing the placeholder "X"

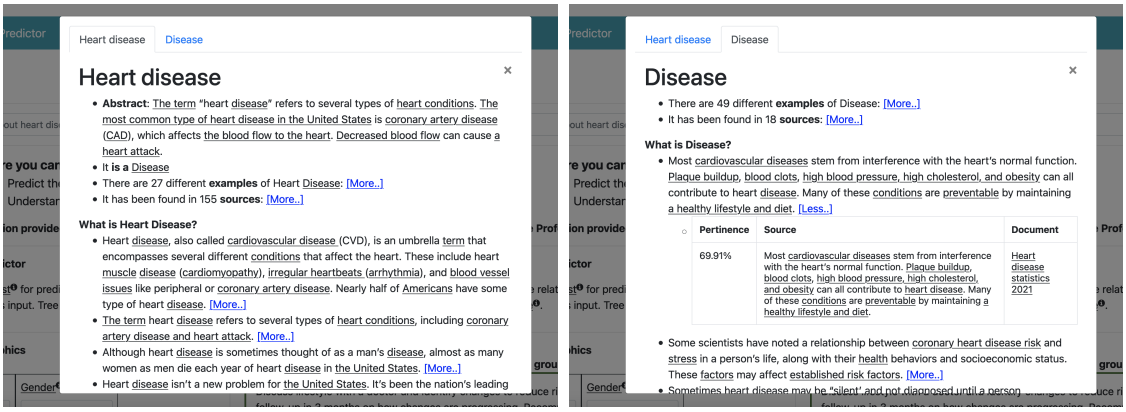


Figure 7.5: ***Overviews generated by YAI4Hu for the heart disease predictor.*** *A screenshot showing how overviews are displayed in YAI4Hu. The first overview (left) is about an explanandum aspect called "Heart Disease", and the second one is about "Disease" (a super-class of "Heart Disease").*

with a label of the aspect to overview. The templates used by YAI4Hu are "`what` is X", "`why` X", "`what` is X for", "`how` is X", "`who` is X", "`where` is X" and "`when` is X". Also, the how-why narrator uses predefined question templates, but these are only: "`why` X" and "`how` is X".

7.3 User Study Design: Quizzes and Questionnaires to Quantify Usability

To test Hypotheses 1 and 3 (cf. Sections 3.2 and 5.4, respectively), it may be sufficient to perform one or more user studies on the two explananda described in Section 7.1, collecting the usability scores of the four explanatory tools presented in Section 7.2. As discussed in Section 3.2, usability scores can be used as a proxy for measuring user-centrality in terms of *effectiveness*, *efficiency*, and *satisfaction*. Thus, we designed a user study that follows a between-subjects experimental design so that each participant can test and evaluate both explananda (starting with the credit approval system, the simplest) but is randomly assigned to only one explanatory instrument (the normal XAI-based explainer, the two-level explainer, the how-why, or YAI4Hu) and not to multiple instruments.

The effectiveness and efficiency of explanations are measured by giv-

ing the explanations to the participants of the user studies and asking them questions to see whether the given explanations helped them to understand the explananda. To this end, two domain-specific multiple-choice quizzes (one per explanandum) are used, each consisting of questions representing plausible information goals for the system users. Being impossible and unfeasible to identify all the possible questions a real user would ask to reach its goals, we decided to select only a few representative questions for the sake of the study. In particular, it appears from preliminary studies, as the one by Liao et al. [124], that users are interested in asking a variety of different questions about an AI system, pointing to complex and heterogeneous needs for explainability that go beyond the output of a single XAI. Therefore, we picked different types of questions, with different complexity and archetypes, using as reference for each explanandum the main user objectives discussed in Section 7.2.

The heart disease predictor and the credit approval system have different but well-defined purposes. Most importantly, many of the questions were selected so that:

- Providing the correct answers would require the exploration of at least 2 or 3 different overviews, with the how-why narrator and YAI4Hu;
- The answers reachable via open-ended question-answering (in YAI4Hu) are not always as accurate as required (with the correct ones not ranked first) or are wrong (i.e., questions 1 and 6 of Table 7.1 and questions 1, 2 and 3 of Table 7.2).

We selected 4 to 8 plausible answers for each question, of which only one was (the most) correct. One of the (wrong) answers was always "I do not know".

Importantly, we decided to use the number of correct answers as *attention check*, discarding all participants with less than one correct answer per quiz. That is because people failing to answer at least one question are likely to be answering (more or less) randomly/nonsensically, paying no attention to the task. Indeed, it is sufficient to read the initial explanations provided by the systems to answer some of the questions (e.g., the first one of the credit approval system).

The questions selected for the quiz on the heart disease predictor are given in Table 7.2. Instead, the questions selected for the quiz on the credit approval system are given in Table 7.1, where the last two questions are about the specific technology used by the system. In fact, in this specific

Table 7.1: ***Quiz of the credit approval system.*** *This table contains the quiz used to evaluate a tool's effectiveness in explaining the credit approval system. Column "QB Type" is the type of question according to the taxonomy of user needs for explainability proposed by Liao et al. [124] in their XAI Question Bank (QB). Column "Steps" indicates the minimum number of steps (in terms of links to click, overviews to open or questions to pose) required by each explanatory tool ("XAI" is the XAI-based explainer, "HWN" is the how-why narrator, and 2EC is the two-level explainer) to provide the correct answer. Negative steps means that the correct answer cannot be found, while 0 steps mean that the answer is immediately available in the initial explanans. Instead, "no OQA" means that open-ended question-answering does not answer the question correctly. Column "Archetype" indicates which archetypes represent the question. Many questions are polyvalent in that they can be rewritten using different archetypes.*

Question	**Archetype**	**QB Type** [124]	**Steps**			
			XAI	**2EC**	**HWN**	**YAI4Hu**
What did the Credit Approval System decide for Mary's application?	`what`, `how`	Output	0	0	0	0
What is an inquiry (in this context)?	`what`	Terminological	-1	1	1	1
What type of inquiries can affect Mary's score, the hard or the soft ones?	`what`, `how`	How (global)	-1	1	1	1
What is an example of hard inquiry?	`what`	Terminological	-1	1	-1	1
How can an account become delinquent?	`how`, `why`	How to be that	-1	1	1	1
Which specific process was used by the Bank to automatically decide whether to assign the loan?	`what`, `how`	How (global)	0	0	0	0 (no OQA)
What are the known issues of the specific technology used by the Bank (to automatically predict Mary's risk performance and to suggest avenues for improvement)?	`what`, `why`	Performance	-1	1	1	1 (no OQA)

context, the data subject (the loan applicant) should be aware of the technological limitations and issues of the automated decision maker (i.e., the credit approval system), as pointed out in Section 1.1.

At the end of an effectiveness quiz, answers are automatically scored as correct (score 1) or not (score 0), and the resulting scores are added together to form the effectiveness score. For example, for the question "What did the Credit Approval System decide for Mary's application?", the correct answer is "It was rejected", and some of the wrong answers are "Nothing" or "I do not know".

Intuitively, the heart disease predictor is a much more complex ex-

Table 7.2: ***Quiz of the heart disease predictor.*** *This table contains the quiz used for evaluating the explainers of the heart disease predictor. For further details about interpreting this table, read the caption of Table 7.1.*

Question	Archetype	QB Type [124]	Steps			
			XAI	2EC	HWN	YAI4Hu
What are the most important factors leading that patient to medium risk of heart disease?	what, why	Why	0	0	0	0 (no OQA)
What is the easiest thing the patient could do to change his heart disease risk from medium to low?	what, how	How to be that	0	0	0	0 (no OQA)
According to the predictor, what serum cholesterol level is needed to shift the heart disease risk from medium to high?	what, how	How to be that	0	0	0	0 (no OQA)
How could the patient avoid raising bad cholesterol, preventing his heart disease risk from shifting from medium to high?	how	How to be that	-1	1	2	2
What tests can be done to measure bad cholesterol levels in the blood?	what, how	Input	-1	1	-1	1
What are the risks of high cholesterol?	what, why not	Output, What if	-1	1	2	1
What is LDL?	what	Terminological	-1	1	2	1
What is Serum Cholesterol?	what	Terminological	-1	1	1	1
What types of chest pain are typical of heart disease?	what, how	How to still be this	-1	1	1	1
What is the most common type of heart disease in the USA?	what	Social	-1	1	1	1
What are the causes of angina?	what, why	Why	-1	1	2	1
What kind of chest pain do you feel with angina?	what, how	Terminological	-1	1	1	1
What are the effects of high blood pressure?	what, why not	Why not, Follow-up	-1	1	1	1
What are the symptoms of high blood pressure?	what, why, how	How (global), Input	-1	1	1	1
What are the effects of smoking on the cardiovascular system?	what, why not	Why not, Follow-up	-1	1	3	1
How can the patient increase his heart rate?	how	How to be that	-1	1	3	1
How can the patient try to prevent a stroke?	how	How to be that	-1	1	3	2
What is a Thallium stress test?	what, why	Terminological	-1	1	3	1

planandum with many more resources and questions to answer. So, we kept the size of the two quizzes proportional to the complexity and richness of the explananda. However, this pushed some participants not to test the heart disease predictor because too burdensome in terms of the minimum time required to complete the quiz.

Furthermore, in order to better understand the relevance to XAI of the questions considered for this user study, we aligned each question to the types of *explainability needs* identified by Liao et al. [124] in their XAI Question Bank.

Though, it could be argued that these questions were arbitrarily chosen and might not be of interest to every explainee. Moreover, the answers to these questions might not always be correctly given by the explanatory tools (i.e., for the adopted AI and XAI providing approximate or wrong

information). However, with these quizzes, we can analyse the quality of the considered explanatory tools and their presentation logic on a large variety of different explainability needs (i.e., almost all of those mentioned in [124], as shown in Tables 7.1 and 7.2). This is regardless of the correctness of the explainable information used for generating the explanations and without making assumptions about the background knowledge of the explainee[11].

In addition to the two aforementioned quizzes, during the experiment all participants are also asked to complete a SUS questionnaire (see Section 3.2) per explanandum (used to measure satisfaction), a (short) Need for Cognition Score (NCS) questionnaire, and to optionally provide some qualitative feedback in the form of a comment.

NCS [39, 56] is a user characteristic that refers to the user's tendency to engage in and enjoy thinking. NCS has become influential across social and medical sciences, and it is not new to the human-computer interaction community either [137]. According to de Holanda Coelho et al. [56], NCS can be measured through a specific questionnaire of 6 items, which responses are given on a 5-point scale (1 = extremely uncharacteristic of the user; 5 = extremely characteristic of the user). NCS scores are computed by summing the given points (from 1 to 5 for questions 1,2,5, and 6; from -5 to -1 for questions 3 and 4) for each questionnaire item.

The relevance of NCS in our study stems from the potential variability in the usability of an explanatory tool across people with low, normal, or high NCS. It is plausible that individuals with a high NCS, often characterized by their dedicated and focused approach, may be more willing and able to navigate a comprehensive explanatory tool like the two-level explainer. Conversely, those with a low NCS might shy away from tasks that require considerable cognitive effort, such as understanding a complex explanandum, potentially leading to dissatisfaction with any explanatory tool that demands more than minimal engagement. Given these considerations, it is important to assess the usability of a user-centred explanatory tool across the entire spectrum of NCS. The term "normal" here denotes scores that are neither extremely high nor low. Moving forward, we aim to examine the correlation between NCS and the effectiveness or satisfaction scores to provide a more representative evaluation. This approach can offer valuable insights into how users with different cognitive inclinations interact with the tool, and help us understand whether individuals with higher NCS, who

[11]The only assumption that is made is that explainees can read and understand English.

might mirror the dedication of actual credit officers, report higher satisfaction or effectiveness scores.

To find out whether a person has a "normal" NCS, it should be enough to calculate the interquartile range of a sufficient number of NCS scores. NCS scores that fall within the interquartile range can be called "normal" because the interquartile range is the range of scores that are neither too high nor too low. The interquartile range is meaningful when no assumptions can be made about the distribution of scores in the population of users participating in the study.

Finally, during the user study, it should be clear to all participants what their expected goal is (e.g., to obtain an explanation; to complete a quiz with the best possible score). So that satisfaction can be adequately measured as the system's ability to meet the user's goals. This could be achieved by explicitly and immediately informing the participants when they fail or succeed in achieving the intended goals so that the user can know whether he or she has indeed acquired the explanation he or she was looking for, thus being satisfied. Consequently, participants should not be paid or rewarded to correctly measure satisfaction in this context. If participants participate in the study only to be paid or rewarded, their objective would be to obtain money as quickly as possible rather than an actual explanation.

7.4 Experiments and Results Discussion

To verify Hypotheses 1 and 3 (cf. Sections 3.2 and 5.4, respectively), we performed the user study presented in Section 7.3 on more than 190 different human subjects coming from two different pools. More specifically, we performed two variations of the same study, one for each user group.

The first user study involved 89 unique participants amongst university students of the following courses of study[12]: bachelor's degree in computer science or in management for informatics (students between 19 and 23 years old); master's degree in digital humanities or in artificial intelligence (students between 21 and 25 years old). In the end, there were approximately 20 participants per explainer.

Participants were told that completing the quizzes and questionnaires (on both the explananda) would have taken an average time that varies from

[12] All study courses took place at the University of Bologna, and only the master's degrees were international, i.e., with English teachings and students from countries other than Italy.

10 to 25 minutes and to use a desktop or laptop because the explanatory tools were not designed for touchscreens or small devices. They were also informed, in a simple and very concise way, that the goal of the survey was to understand which explanatory mechanism (amongst many) is the best, without going into further details. Therefore, participants knew that other versions of the explanatory tool were available and that other users may have received a different one.

Furthermore, participants were explicitly asked to use only the information reachable from within the systems (i.e., by following the hyperlinks there). In other terms, they were clearly instructed not to use Google or other external tools. Participants were also:

- Instructed to click on "I don't know" in case they did not know an answer;
- Informed that there is only one correct answer for each question, and when multiple answers seem to be correct, only the most precise is considered to be the correct one;
- Noticed when a wrong answer was given, showing them the correct one to make them aware of their success or failure in reaching a goal.

Questions were shown in order, one by one, separately, and answers were randomly shuffled. For the credit approval system, we got 89 participants, as shown in Table 7.3, while for the heart disease predictor, we got 70 participants. Eventually, we collected 70 valid participants taking the NCS test for the credit approval system and 48 for the heart disease predictor. As shown in Figure 7.6, the resulting NCS median score[14] was 8 with a lower quartile of 5 and an upper quartile of 11. Therefore participants with a "normal" NCS s were those with $5 \leq s \leq 11$. The mean NCS was 7.55.

As shown in Figures 7.7 and 7.8, YAI4Hu is visibly the most effective and satisfactory explanatory tool in both the explananda, followed by the how-why narrator, while the XAI-based explainer seems to be the worst

[13]The statistics shown in Table 7.3 for the heart disease predictor are slightly different from those presented in [190] because we identified and corrected a bug that caused the script to identify some respondents as having failed the attention check.

[14]In [189], the median score mentioned is different because the user study considers a smaller number of participants.

[15]The results shown in Figure 7.8 for the heart disease predictor are slightly different from those presented in [190] for the same reason as Table 7.3.

Table 7.3: ***Statistics on the participants to the user study.***[13] *This table shows the number of participants adhering to the user study for both the heart disease predictor (HD) and the credit approval system (CA) and each explanatory approach: the normal XAI-based explainer (XAI, for short), the two-level explainer (2EC, for short), the how-why narrator (HWN, for short) and YAI4Hu. The first column ("Respondents") shows the total number of respondents. The second column ("Check") shows only the number of respondents that passed the attention check. The third column ("Check+NCS") shows only the number of respondents that passed the attention check and completed the NCS questionnaire. The box-plots of Figures 7.6 and 7.7 consider only the respondents of the third column, while the box-plot of Figure 7.8 also considers the respondents of the second column.*

		Respondents	Check	Check+NCS
CA	XAI	21	20	16
	2EC	21	21	19
	HWN	18	18	15
	YAI4Hu	29	26	20
HD	XAI	15	14	8
	2EC	17	16	16
	HWN	17	16	12
	YAI4Hu	21	20	12

overall, followed by the two-level explainer. This is true, even if YAI4Hu does not perfectly implement the SAGE-ARS model, i.e., by not implementing sophisticated mechanisms for *adaptivity* (the A of SAGE) or by implementing in a naive way the *relevance* heuristic.

The difference in usability between participants with normal and those with non-normal NCS can be noted by observing the differences between Figures 7.7 and 7.8. As expected, there is a decrease in satisfaction for the more user-centred tools and an increase in effectiveness for the two-level explainer (at least with regard to the heart disease predictor). Only people with a high NCS are more effective with a two-level explainer. As shown in Figures 7.7 and 7.8, YAI4Hu is visibly the most effective and satisfactory explanatory tool in both the explananda, followed by the how-why narrator, while the XAI-based explainer seems to be the worst overall, followed by the two-level explainer. The difference in usability between

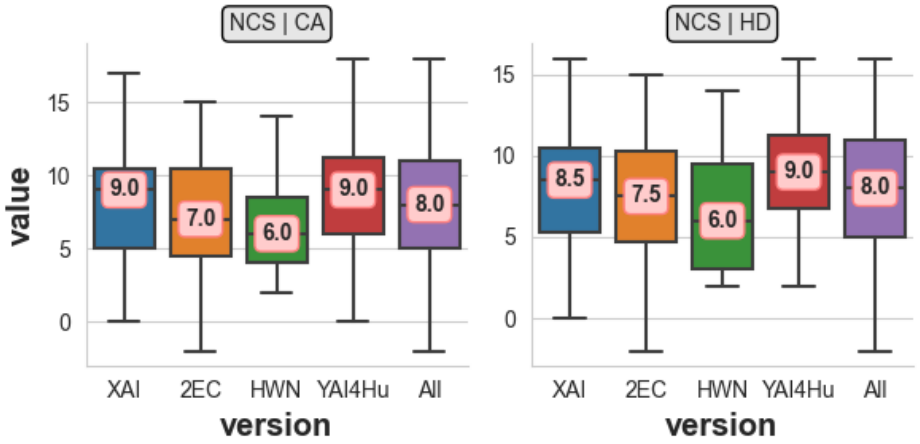


Figure 7.6: ***NCS scores of those participants that passed the attention check for both the credit approval system (CA) and the heart disease predictor (HD).*** *Results are shown as box plots (25th, 50th, 75th percentile, and whiskers covering all data and outliers). The numerical value of the medians is shown inside pink boxes. The results for the normal XAI explainer are in blue. For the two-level explainer, they are in orange. For the how-why narrator, they are in green. For YAI4Hu, they are in red. For all explainers together, they are in purple.*

participants with normal and those with non-normal NCS can be noted by observing the differences between Figures 7.7 and 7.8. As expected, there is a decrease in satisfaction for the more user-centred tools and an increase in effectiveness for the two-level explainer (at least concerning the heart disease predictor). Only people with a high NCS are more effective with overwhelming explanatory closures.

These trends are in line with our expectation, derived from Hypothesis 1 (cf. Section 3.2) that greater illocutionary force and goal-orientedness imply greater usability. On the one hand, the normal XAI-based explainer has the smallest degree of explanatory illocution. In contrast, both the two-level explainer and YAI4Hu have the greatest because their contents can be used to answer more implicit questions as well as all quiz questions (as shown in Tables 7.1 and 7.2). Although slightly smaller, the how-why narrator also has a similar illocutionary force as the two-level explainer and YAI4Hu. On the other hand, YAI4Hu has the greatest degree of goal-orientedness (because it implements both open-ended question-answering and overview-

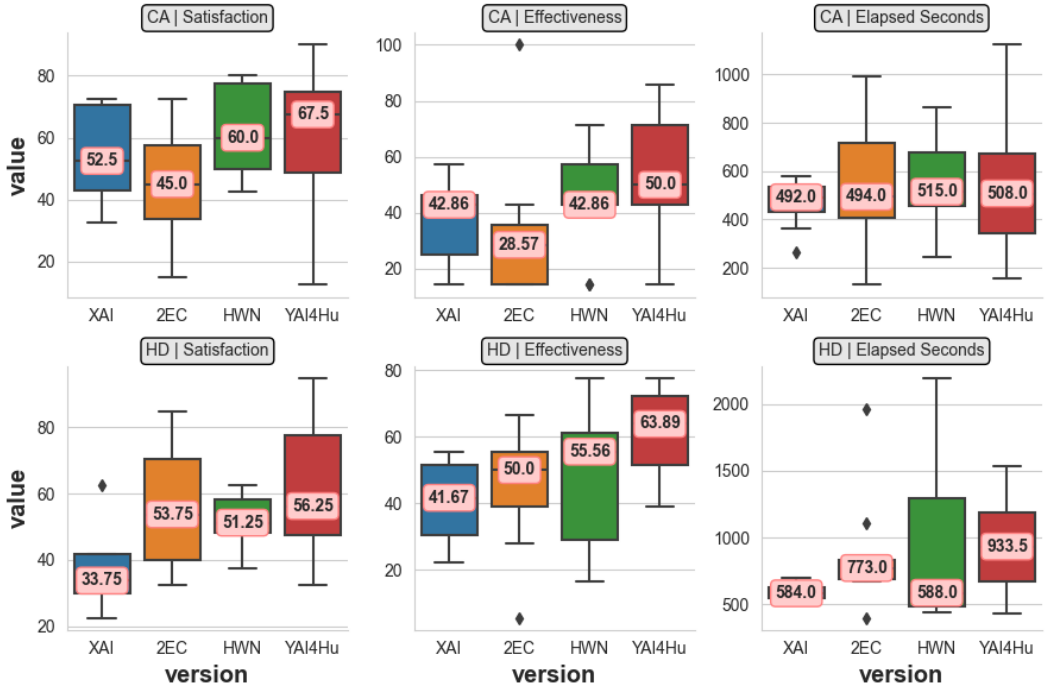


Figure 7.7: ***Usability scores - All vs YAI4Hu - Normal NCS.*** *Only participants with a normal NCS are considered in this figure. Results are shown as box plots (25th, 50th, 75th percentile, and whiskers covering all data and outliers). The numerical value of the medians is shown inside pink boxes. The 1st row is for the heart disease predictor (HD), while the 2nd for the credit approval system (CA). Satisfaction is shown in the 1st column, effectiveness in the 2nd, and elapsed seconds in the 3rd. In this picture, we abbreviate the XAI-based explainer as XAI, the two-level explainer as 2EC and the how-why Narrator as HWN. Effectiveness scores are normalised in* $[0, 100]$.

ing), followed by the how-why narrator, the two-level explainer and the normal XAI-based explainer.

Furthermore, the obtained results highlight a *good correlation* between objective (i.e., effectiveness) and subjective (i.e., satisfaction) metrics in both the explananda, even if it is more evident in the credit approval system. We believe that this difference between the results of the credit approval system and the heart disease predictor is because the latter is much more complex, considering that no participant was able to obtain effectiveness

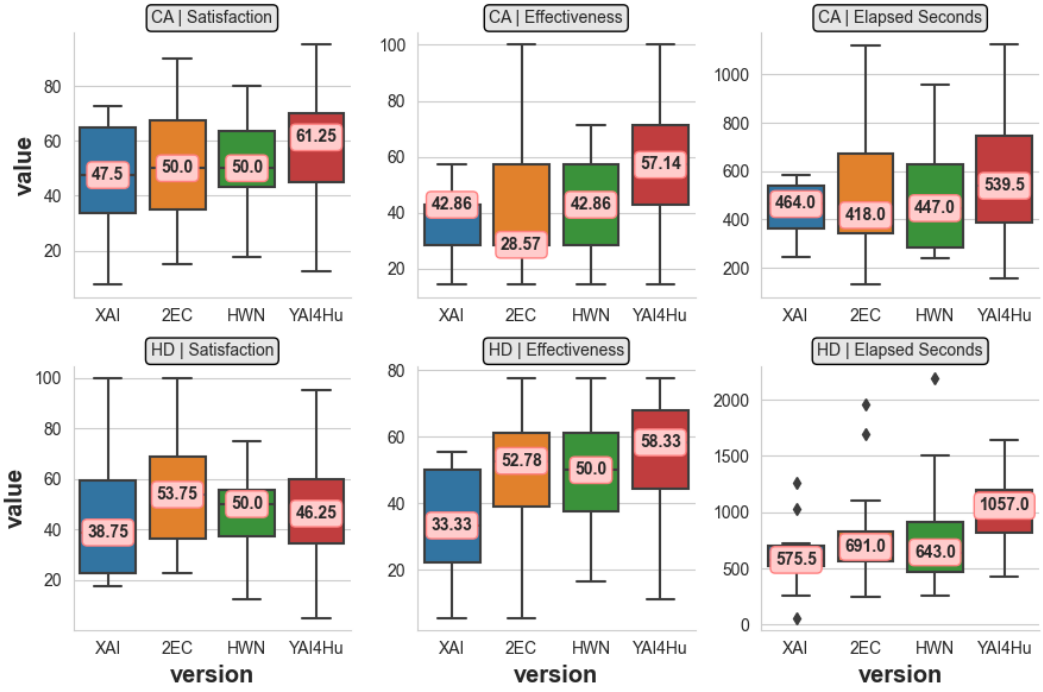


Figure 7.8: ***Usability scores - All vs YAI4Hu - Any NCS.***[15] *In this figure, participants with any NCS are considered (i.e., the respondents of the second column in Table 7.3), not just those with a NCS within the interquartile range. For more details about how to read this figure, see Figure 7.7. Effectiveness scores are normalised in* $[0, 100]$.

scores higher than 80%. Indeed, the average number of minimally required steps (to reach the information containing an answer) is higher in the heart disease predictor, as shown in Table 7.2. This may suggest that the intrinsic complexity of the explanandum influences satisfaction with an explanatory process in a *different* way than effectiveness.

Nonetheless, the results indicate that Hypothesis 3 (cf. Section 5.4) is also correct, as not all decompositions of the explanatory space are maximally helpful for a generic human subject. Both the how-why narrator and YAI4Hu implement the ARS heuristics, but YAI4Hu outperforms the how-why narrator in terms of effectiveness and satisfaction. The main difference between the two explanatory tools is that the how-why narrator is less *grounded* (the G of SAGE), not fully implementing the explanatory process as an illocutionary act of answering questions.

Similarly, the how-why narrator outperforms a two-level explainer that is even less *grounded*, implementing neither *relevance* nor *simplicity*. This indicates that, as expected, an overwhelming and superficial decomposition of the explanatory space may not be helpful for a human subject. Thus, although several decompositions of the explanatory space can be found, not all are equally useful and explanatory, suggesting that a full implementation of both the ARS heuristics and SAGE commands may be necessary to explain effectively.

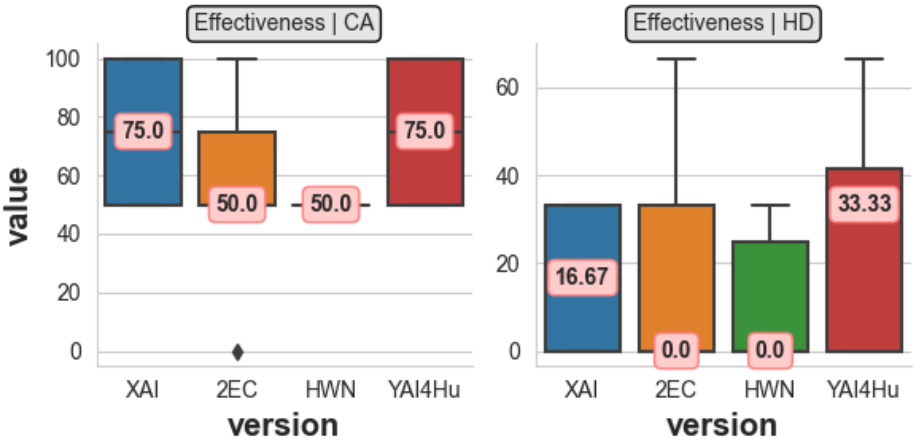


Figure 7.9: ***Effectiveness scores - All vs YAI4Hu - Normal NCS on questions that can be answered with the information provided by the XAI-based explainer.*** *The questions that can be answered with the information provided by the XAI-based explainer are shown in Tables 7.2 and 7.1. For more details about interpreting this figure, read the caption of Figure 7.7. Effectiveness scores are normalised in* $[0, 100]$.

Specifically, these experiments show that illocution can lead to a significant increment in effectiveness on both the considered explananda and that this increment is improved by goal-orientedness. Even though the two-level explainer can technically answer all the questions in the quiz (having a greater illocutionary force), it still performed worse than the how-why narrator and YAI4Hu. Moreover, it seems that implementing explanatory illocution without goal-orientedness (as the two-level explainer does) or goal-orientedness without enough illocutionary force (as the how-why narrator) might be even harmful. This is shown in Figure 7.9, where the performance of the XAI-based explainer is better than that of the two-level

explainer and the how-why narrator.

The performance of YAI4Hu is better (in the case of the heart disease predictor) or equal (in the case of the credit approval system) to that of the normal XAI-based explainer, in terms of median effectiveness score, on the explanations provided by the XAI-based explainer (questions 1, 2 and 3 in the heart disease predictor quiz). This further indicates that explanations can be more than a textual or visual presentation of the information provided by a XAI.

These insights are also supported by the **qualitative feedback** provided by participants. Specifically, the feedback was:

- **Overall negative for the two-level explainer**, i.e., "*I did not understand the website's purpose for this quiz, as I did not feel it helped anything. If the point was to use the available links at the site, there were too many of them, so it was no longer useful*";
- **Overall neutral for the how-why narrator**. Most of the comments were like "*I have no comment*", except a few negative ones, i.e., "*Too long, too difficult, strange way to ask the question... it was not very clear!*";
- **Overall positive for YAI4Hu**, i.e., "*This time, the accuracy was surprisingly great: most of the time, the correct answer was the first to be given. However, in a couple of cases, the answer wasn't even among the ones given (and the system still counted them as sufficient). I noticed that this happens especially with more general questions, such as "what is ..." and therefore had to click on the name to know the right answer, while more specific questions (such as "what causes..." or "who suffers most...") were easier for the system to find*". Though some suggestions for improvements were also given, i.e., "*The given information for each answer was a lot, and not always the answer I was looking for was among the first; also, there could have been more possible answers with for the same question, but separated in the list*".

We performed a few one-sided Mann-Whitney U-tests [128] (a non-parametric version of the t-test for independent samples) on the global (between-subjects) scores. We did it to thoroughly verify Hypotheses 1 and 3, discarding the possibility that the outcomes are the result of luck. The results shown in Figure 7.8 indicate that the distribution of scores is

skewed, with medians usually closer to one of the other quartiles. Therefore, due to the limited number of samples, we chose not to make assumptions of parametrisation in the data[16] collected throughout the user study, which forced us to rely on non-parametric tests (i.e., Mann-Whitney).

Table 7.4: ***One-sided Mann-Whitney U-tests - All vs All - Any NCS.***[17] *This table shows the results of one-sided Mann-Whitney statistical tests (without Bonferroni correction) comparing each explainer involved in the experiment without filtering the NCS scores. The columns indicate the alternative hypotheses (U is the Mann-Whitney statistics, while p is the p-value). Instead, the rows indicate that the statistical tests were performed on the effectiveness or satisfaction scores of the credit approval system (CA) or the heart disease predictor (HD). P-values lower than .05 are shown in bold and considered statistically significant.*

		YAI4Hu > HWN	YAI4Hu > 2EC	YAI4Hu > XAI	HWN > 2EC	HWN > XAI	2EC > XAI
CA	Satisfaction	U=184.5 *p*=.12	U=217 *p*=.11	U=182 *p*=.06	U=180.5 *p*=.41	U=155 *p*=.31	U=190.5 *p*=.4
	Effectiveness	**U=145.5** ***p*=.01**	**U=159** ***p*=.006**	**U=111.5** ***p*=.0007**	U=173.5 *p*=.33	U=146 *p*=.22	U=201.5 *p*=.52
HD	Satisfaction	U=166 *p*=.58	U=184 *p*=.78	U=117 *p*=.21	U=150 *p*=.8	U=90 *p*=.18	U=77.5 *p*=.07
	Effectiveness	U=116.5 *p*=.08	U=121.5 *p*=.11	**U=45** ***p*=.0004**	U=143 *p*=.72	**U=70** ***p*=.04**	**U=50.5** ***p*=.005**

The results of the statistical tests (without Bonferroni correction) on the effectiveness and satisfaction scores shown in Figure 7.8 are given in Table 7.4. In particular, assuming that a p-value lower than .05 is enough for asserting statistical significance[18], we have that YAI4Hu is significantly more effective than the how-why narrator, the two-level explainer and the normal XAI-based explainer on the credit approval system. We also have that the normal XAI-based explainer is significantly less effective than all the other explainers on the heart disease predictor.

Considering that we are doing three multiple statistical tests per score, the chances of having a test that falsely results as expected increase. Some

[16] The anonymised data is available at `https://github.com/Francesco-Sovrano/YAI4Hu`, for reproducibility purposes.

[17] Differently from [190], in Table 7.4 we show statistical tests without filters on the NCS of participants. This way, tests should be more reliable since filtering on NCS reduces the number of data samples.

[18] Note that a p-value greater than or equal to .05 does not imply that the null hypothesis is valid.

statistical tools that are used in this case to reduce the chance of a type I error (false positive) are: the Bonferroni correction, the Holm–Bonferroni method, or the Dunn–Šidák correction. Though these tools are known to increase type II errors (false negatives) [7]. Regardless, if we used a Bonferroni or Dunn correction to adjust for three multiple comparisons per score, then the minimum p-value for claiming a statistically significant result would not be $.05$ but something close to $.016$. However, even with these corrections, all the claims of statistical significance would still hold, except for the one of the how-why narrator being more effective than the normal XAI-based explainer on the heart disease predictor.

In order to further validate our results from a statistical point of view, we repeated the same experiment with 103 new participants (57 males, 44 females, two unknown, aged between 18 and 55, resident in the UK, US or Ireland), recruited through the online platform Prolific [150]. Considering that the participants were paid £7.56 per hour, we decided to repeat the between-subjects user study presented in Section 7.3 on the credit approval system only. We did it without measuring satisfaction and comparing only the normal XAI-based explainer with YAI4Hu without open-ended question-answering. Once again, results showed that the global (between-subjects) effectiveness of YAI4Hu is significantly greater than the normal XAI-based explainer[19] (U=931, p=.03), even without open-ended question-answering.

However, these experiments have some limitations to highlight. Firstly, our evaluation of the explanatory mechanisms is intertwined with the user interface, making it difficult to understand the primary sources of usability problems. Secondly, the algorithm pipeline relies on several heuristics and approximations that may hinder the explanatory systems' usability. For instance, the answer retrieval mechanism could be better and, on several occasions, fails to provide the best answer, as pointed out by several users. Therefore, in the following chapters, we will discuss how to implement explainability evaluation mechanisms independent of the user interface and strategies to improve answer retrievers' performance on technical documentation even without expensive (training) procedures and datasets.

One strength but also another possible limitation of YAI4Hu is that we evaluated it only on generic laypersons, without making any assumption on the background knowledge of the explainee. Though, tools such as YAI4Hu might be less effective with different types of users (e.g., field

[19]For more details about this second experiment, read [188].

experts and regulators). That is because the *relevance* of information is different for expert users. For example, they might be more interested in having more specific and complex information first. However, open-ended question-answering might easily overcome the issue, suggesting that a combination of overviewing with open-ended questioning is needed. Nevertheless, as future work, more intelligent and more adaptive strategies for aspect overviewing might be designed to improve the user experience of an expert explainee on YAI4Hu.

CHAPTER 8

Objective Quantification of Textual Explainability: an Empirical Analysis of the DoXpy Algorithm

In a recent attempt to capture the "legal requirements on explainability in machine learning", Bibal et al. [22] identified four primary explainability requisites for Business-to-Consumer and Business-to-Business, analysing the provisions of European law. Specifically, in these cases, explanations about a solely-automated decision-making system should at least provide information about the following:

- The main features used in a decision taken by the AI;
- All features processed by the AI;
- The specific decision taken by the AI;
- The underlying logical model followed by the AI.

Interestingly, if Hypothesis 2 (cf. Chapter 4) is valid, then it would be possible to use the work of Bibal et al. [22] to objectively quantify through DoX (cf. Chapter 4) how much of the information required by the law is explained by an AI system.

To test Hypothesis 2, we performed two experiments, both aimed at showing that explainability changes following DoX. To conduct the experiments, we considered the XAI-based systems and explanatory tools presented in Chapter 7. To do so, we used the answer retrieval mechanism described in Chapter 6 to implement DoXpy, an algorithm capable of estimating the DoX of any arbitrary piece of textual information.

The first experiment followed a *direct* approach, comparing the DoX of the XAI-based systems with their non-explainable counterpart. This approach is said to be direct because the amount of explainability of an XAI-based system is, by design, clearly and explicitly dependent on the output of the underlying XAI. Therefore, by filtering away the output of the XAI, the overall system can be forced to be not explainable enough by construction.

On the contrary, the second experiment followed an *indirect* approach, analysing the expected effects of explainability on the explainees. If Hypothesis 2 is correct, the lower DoX is, the fewer explanations can be extracted, the less effective (as per ISO 9241-210) the explainee is likely to be in reaching those explanatory goals that are not covered by the explanations. To show this, we borrowed the results of the user studies used to evaluate YAI4Hu (cf. Chapter 7), studying how DoX correlates with the effectiveness scores measured by the user studies.

In this chapter, we will explain the details of the previous experiments (also discussing how we implemented DoXpy), which pertinence functions p and threshold t we considered for computing the DoX scores, and how we identified the set A of explanandum aspects.

8.1 The Pipeline of DoXpy

Throughout this section, we will explain how to use existing algorithms for answer retrieval and information extraction to implement DoXpy, an algorithm for computing DoX. For reproducibility purposes, we publish the DoXpy source code at `https://github.com/Francesco-Sovrano/DoXpy`.

Given Definition 6 (cf. Section 4.1.1), we argue that it is possible to write an algorithm that can approximately quantify the Degree of Explain-

ability of information representable with *natural language* (e.g., English) by adapting existing technology for question-answering. According to Definition 5, in order to implement an algorithm capable of computing the (average) DoX of Φ, we need to:

- Define a set A of *explanandum aspects*;
- Identify the set of all possible *archetypes* Q;
- Define a mechanism to identify the set D of *details* contained in Φ and the subset D_a for every $a \in A$;
- Define the *question-answering process*: the function p to compute the pertinence of an individual detail d to an archetypal question q_a.

Notably, the set of aspects A is task-dependent and must be defined for each explanandum (e.g., manually listing all aspects or automatically extracting the list of aspects from a textual description of the explanation with a tokenizer). Instead, the set of archetypes Q, the pertinence function p, and the mechanism for extracting D and D_a from Φ *may always be the same* for all explananda. For example, as Q, it may be sufficient to consider the archetypal questions identified in Section 3.3, being generic and rich enough to capture all elementary discursive units and abstract meaning of any (English) text.

In particular, the set A of explanandum aspects is a collection of (lemmatised) words, and it can be different from the set I of aspects explained by Φ. What is of utmost importance for a Φ to be a good explanandum support material is that $A \subseteq I$.

A detail d is a snippet of text called *information unit*, a relatively small sequence of words about one or more aspects (i.e., a sub-set of I) that is usually extracted from a more complex information bundle (i.e., a paragraph, a sentence). In other terms, these details should carry enough information to describe different parts of an aspect (possibly connected to many other aspects). So, we can use them to answer some (archetypal) questions about an $a \in A$ and to correctly estimate a *level of detail*, as required by Definition 6 (cf. Section 4.1.1).

Considering the characteristics of D and I mentioned above, the most natural representation of them is a (knowledge) graph. A graph is a set of nodes I connected by a set of edges D. Therefore, we believe that the simplest way to identify the set of details D may be to extract a graph of

information units from Φ on which efficient question-answering could be performed.

Thus, an approach such as the one described in Chapter 6 for (archetypal) question-answering could be suitable for our purposes. It would allow the identification of meaningful information units and also suggest a mechanism for estimating *pertinence* by extracting from Φ a graph of D and I designed for answer retrieval. Importantly, as information units, YAI4Hu uses grammatical clauses (meaningful decompositions of grammatical dependency trees) to ensure that the units represent the smallest granularity of information.

As a consequence, using this type of information units for DoX guarantees:

- A disentanglement of complex information bundles into the most simple units, to correctly estimate the *level of detail* covered by the information pieces, as per Definition 6;
- A better identification of duplicate units scattered in the information pieces to avoid an over-estimation of the *level of detail*.

All these properties satisfy the requirements that a good detail $d \in D$ should possess for generating a DoX score. This motivates our decision to use YAI4Hu's answer retrieval algorithm as the main component of the DoXpy pipeline.

YAI4Hu's answer retrieval algorithm consists of a pipeline of AI tools specifically designed to measure the pertinence p of D to a set of (archetypal) questions Q on A. As shown in Figure 6.1, DoXpy's answer retrieval algorithm relies on mechanisms for embedding questions and answers in dense numerical representations so that the cosine similarity between the embedding of a question and that of an answer is a measure of the latter's relevance to the former.

More specifically, let a be the explanandum aspect of a question q_a, $m =< s, t, o >$ be a template-triplet, $d = t(s, o)$ be the natural language representation of m also called information unit, and z the context (i.e., a paragraph, a sentence) from which m was extracted. DoXpy performs answer retrieval by retrieving the set D_a of all the template-triplets about a and selecting amongst the natural language representations d of the retrieved template-triplets those that are likely to be an answer to q_a. The probability that d pertinently answers q_a can be estimated as the similarity between the embedding of $< d, z >$ and the embedding of q_a. Therefore,

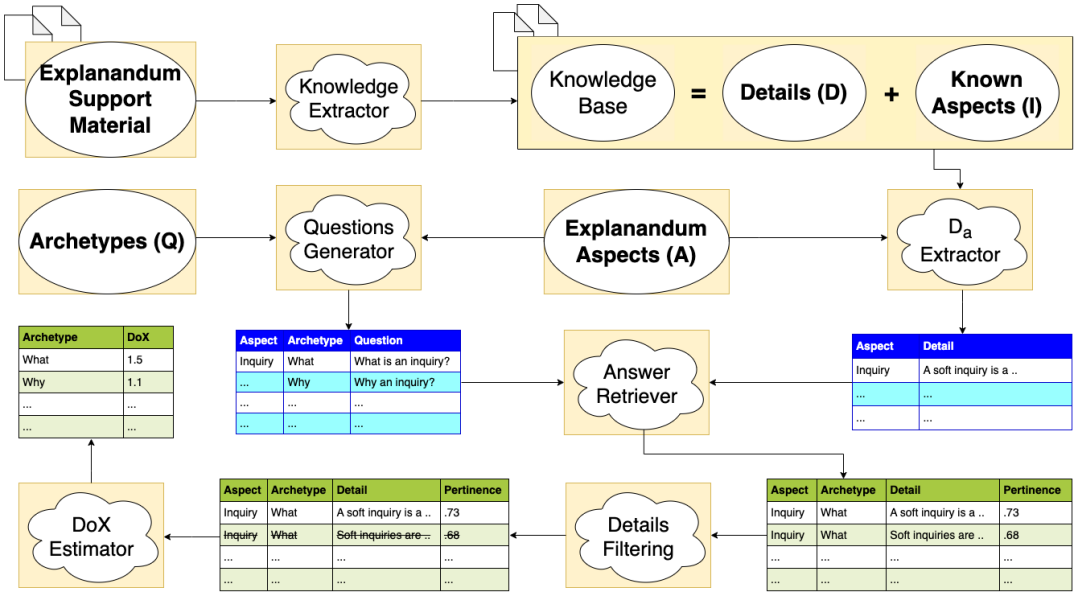


Figure 8.1: ***DoXpy pipeline.*** *The pipeline starts with extracting a graph from the explanandum support material Φ that is then converted into a set of details D. The set of details is combined with the explanandum A and the set of archetypes Q to compute the DoX. To do this, we use some deep language models for answer retrieval.*

in practice, the algorithm can retrieve an unbounded number of details (i.e., answers).

In particular, a detail is said to be redundant (i.e., duplicated) whenever it contains information that answers an archetypal question $q_a \in Q$ in a manner too similar to that of other (more pertinent) details. For example, the detail "*P is the probability of having a heart disease*" is different but similar to "*the score P is the probability of having a disease*". However, the former detail is more precise (it speaks of heart diseases instead of generic diseases) and relevant than the latter in answering the archetypal question "*What is probability P?*". Therefore, to prevent DoXpy from considering two details expressing the same information differently, the second detail must be discarded as redundant. To do this, DoXpy uses the same deep neural networks used for retrieval to compute the similarity between two answers, discarding those with the lowest relevance scores that share a similarity greater than a threshold r.

Consequently, as shown in Figure 8.1, the pipeline of DoXpy consists

of the following four steps. First, a knowledge graph is extracted from the explanandum support material Φ using the algorithm described in Section 6.2, thus defining the set of details D and the set of known aspects I. Secondly, the explanandum aspects A and archetypes Q are used to generate the questions q_a for each $a \in A$ and $q \in Q$, and also to identify all $D_a \subseteq D$. Third, the answer retriever described in Section 6.1 is used to associate a pertinence score with each $d \in D_a$ for each q_a, and (importantly) to identify and filter out duplicate answers. Fourth, the formulas in Section 4.1 are used to aggregate the relevance scores and estimate the (average) DoX without considering duplicate details.

Moreover, according to Definition 6, we need to define a pertinence function p and pick a threshold t to compute the DoX. As previously discussed, we will use as pertinence function p a deep language model for answer retrieval. The point is that many different deep language models exist for this task, i.e., [83, 194, 106], and each one of them has different characteristics producing different pertinence scores. So, which model is the right one for computing the DoX? Can we use any model?

To answer these questions, we decided to study the behaviour of more than one deep language model as pertinence function p. Assuming that these models get good results on state-of-the-art benchmarks for pertinence estimation, we believe that the results of the computation of DoX should be consistent across them. Hence the models we considered are:

- MiniLM: published by [106, 165] and trained on Natural Questions [113], TriviaQA [103], WebQuestions [19], and CuratedTREC [16].
- Multilingual Universal Sentence Encoder: published by [225] and trained on the Stanford Natural Language Inference corpus [28].

In Chapter 7, we conducted experiments on two XAI-based systems and determined that for both of the language models mentioned above, a suitable relevance threshold can be $t = 0.15$ and an appropriate duplication threshold can be $r = 0.85$.

8.2 1st Experiment: Direct Evaluation on Normal XAI-generated Explanations

In Chapter 4, we argued that the degree of explainability of any collection of text (e.g., the output of an XAI-based system) could be measured in terms

of DoX on a set of chosen explanandum aspects. In order to verify this assertion and Hypothesis 2 (cf. Chapter 4), we have to show that there is a strong correlation between DoX and the perceived amount of explainability. To this end, we devised two experiments.

As anticipated at the beginning of Chapter 8, with the first experiment, we measure explainability *directly* to shed more light on how a few changes to the explainability of a system affect the estimated DoX. Specifically, XAI-based systems are considered for this experiment because their amount of explainability is, by design, clearly and explicitly dependent on the output of the underlying XAI. So, by masking the output of the XAI, the overall system can be forced to be less explainable. Hence, this characteristic can be exploited to (at least partially) verify the hypothesis in a straightforward but effective way.

In other words, a XAI-based system is composed of a black-box AI system wrapped by a XAI. So, with this experiment, we compare the DoX of a normal *XAI-based explainer* with that of the same system without the XAI, also called normal *AI-based explainer*. As a result, we expect the (average) DoX of the XAI-based explainer to be higher than its wrapped AI.

For this experiment, we used the XAI-based systems defined in Section 7.1. Therefore, by simply removing the output of the XAI (respectively CEM and TreeSHAP) from these systems, it is possible to obtain the AI-based explainers we need.

In order to compute the (average) DoX of these systems, we take as a set of explanandum aspects those targeted by the credit approval system and the heart disease predictor. More precisely, the main explanandum aspects A targeted by XGBoost [49] and TreeSHAP [126] in the heart disease predictor are five:

- The recommended action for patient X;
- The most important factors Y that contribute to predicting the likelihood of heart disease;
- The likelihood of heart disease;
- The risk R of having a heart disease;
- The contribution of Y to predict the likelihood of heart disease for patient X.

While the main explanandum aspects A targeted by the artificial neural network and CEM [62] in the credit approval system are four:

- The factors F to consider for changing the result;
- The relative importance of factors F in changing the result;
- The risk performance of applicant X;
- The result of the application of applicant X.

Eventually, after properly converting the images produced by the XAI-based explainers to textual explanations, the resulting *explanandum aspects coverage* (i.e., the ratio of $|A \cap I|$ to $|A|$) of both the heart disease predictor and the credit approval system is 100%. In contrast, the aspects coverage of their AI-based explainers is 48% and 43%, respectively.

By calculating the DoX through DoXpy, we obtained the results shown in Table 8.1. As expected, for both the heart disease predictor and the credit approval system, the experiment results indicate that the (average) DoX of all XAI-based explainers is significantly higher than that of AI-based explainers, regardless of the deep language model adopted. Although, we can see that MiniLM and the Universal Sentence Encoder (the two adopted language models) produce comparable but different DoX scores, suggesting that the choice of the pertinence function p can sensibly impact the value of DoX.

In this first experiment, we arbitrarily chose a simple set of explanandum aspects. However, what would happen if we considered different and more complex explananda and explanatory contents? Furthermore, the result of this experiment is based on comparing the DoX of an unexplained system (i.e., the AI-based explainers) with that of a more explainable system, and this is an exceptional and naive case to consider. Therefore, to thoroughly test Hypothesis 2 (cf. Chapter 4), we must understand whether DoX behaves as expected even when explainability is present in different and non-zero quantities. To this end, explainability can be measured *indirectly* by studying the effectiveness of the resulting explanations on human subjects, as shown in Section 8.3.

[1]The numerical values in this table are different from those reported in [191] because we used DoXpy v3.0 instead, which includes several improvements in the information retrieval algorithm that prevent details duplication, as described in Section 8.1.

Table 8.1: ***Results of the 1st experiment on DoXpy***[1]**.** *In this table, DoX and average (Avg) DoX are shown for the credit approval system (CA) and the heart disease predictor (HD). As columns, we have the normal AI-based explainers (AI, for short) and the normal XAI-based explainers (XAI, for short). As rows, we have different explainability estimates using MiniLM (ML) and the Universal Sentence Encoder (TF). For simplicity, for DoX, we show only the primary archetypes.*

		CA		HD	
		AI	XAI	AI	XAI
Avg DoX	ML	0.22	0.83	0.34	0.79
	TF	0.19	0.63	0.24	0.61
DoX	ML	"what": 0.24 "how": 0.23 "who": 0.23 "which": 0.23 "why": 0.23 "whose": 0.22 "when": 0.21 "where": 0.21	"how": 0.87 "which": 0.86 "what": 0.86 "why": 0.84 "when": 0.81 "who": 0.80 "where": 0.78 "whose": 0.77	"why": 0.37 "which": 0.35 "what": 0.34 "how": 0.34 "whose": 0.32 "when": 0.31 "who": 0.31 "where": 0.31	"why": 0.85 "which": 0.84 "what": 0.82 "how": 0.81 "whose": 0.80 "who": 0.77 "when": 0.74 "where": 0.74
	TF	"what": 0.21 "when": 0.20 "which": 0.18 "who": 0.15 "how": 0.14 "where": 0.14 "why": 0.11 "whose": 0.08	"what": 0.71 "when": 0.67 "which": 0.54 "where": 0.51 "how": 0.48 "who": 0.46 "why": 0.41 "whose": 0.31	"what": 0.28 "when": 0.24 "who": 0.17 "where": 0.17 "why": 0.17 "how": 0.16 "which": 0.15 "whose": 0.10	"what": 0.73 "when": 0.57 "how": 0.48 "which": 0.45 "who": 0.44 "where": 0.43 "why": 0.43 "whose": 0.30

8.3 2nd Experiment: A Study of the Effects of Explainability on Human Subjects

This second experiment aims to show whether there is a correlation between DoX and the effects of explainability on human subjects. A higher explainability implies a greater capacity to explain, hence a greater number of explanations. In other words, the lower the DoX, the fewer explanations can be produced, and the less effective the explainer is in explanandum-related tasks. To verify this point, we borrowed the user studies presented in Chapter 7, which involved more than 190 human subjects. Notably, these user studies considered the same explanandum support materials of the first experiment, analysing the effectiveness of explanations given by different

explainers when changing the explanandum support material and the way explanations are presented to the explainee.

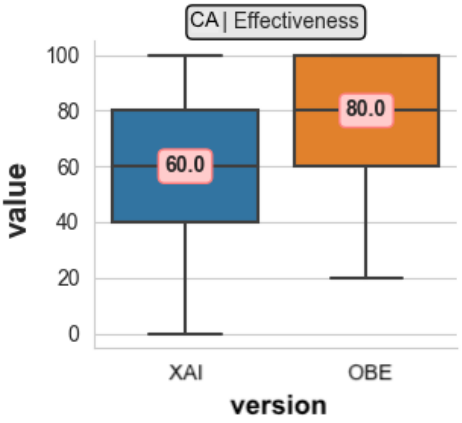


Figure 8.2: ***2nd user study: effectiveness scores on questions that cannot be answered with the information provided by the XAI-based explainer.*** *This figure shows a comparison of the median effectiveness scores obtained on the credit approval system (CA) with the normal XAI-based explainer (XAI; the blue one) and YAI4Hu without open-ended question-answering (called overview-based explainer or OBE for short; the orange one) on those questions whose answer is not provided by the XAI-based explainer. Results are shown as box plots (25th, 50th, 75th percentile, and whiskers covering all data and outliers). The numerical value of the medians is shown inside pink boxes. Differently from [188], here effectiveness scores are normalised in* $[0, 100]$.

Both the results of the (first) user study (which involved 89 participants; cf. Section 7.4) and the (second) user study (which involved 103 participants; cf. Section 7.4) indicate that a more explainable explanandum support material implies an explainer capable of producing more effective explanations. As also shown in Figure 8.2, according to a one-sided Mann-Whitney U-Test, there is enough statistical evidence to claim that the instance of YAI4Hu considered for the second user study is more effective on the credit approval system (U=849.5, p=.007) than the XAI-based explainer on those questions that cannot be answered by the XAI (i.e., questions number 2, 3, 4, 5 and 7 in Table 7.1).

Moreover, as shown in Figure 8.3, the same can be said for the heart disease predictor in the first user study. As expected, also in this case,

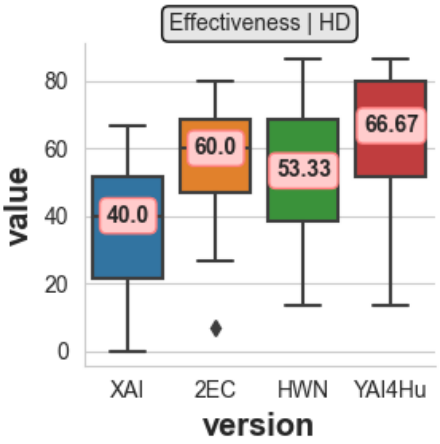


Figure 8.3: ***1st user study: effectiveness scores on questions that cannot be answered with the information provided by the XAI-based explainer.*** *Comparison of the results achieved on the heart disease predictor (HD) with the normal XAI-based explainer and the other explainers, only on those questions whose aspects are not covered by the information presented by the XAI, without filters on NCS scores. The other explainers are the two-layered explainer (2EC), the how-why explainer (HWN) and YAI4Hu. For more details about interpreting this figure, read the caption of Figure 7.7.*

we see the median effectiveness score of the normal XAI-based explainer being significantly lower than the other explainers on the questions that the XAI cannot answer (i.e., the questions with negative steps in Table 7.2). More precisely, according to some one-sided Mann-Whitney U-tests, there is enough statistical evidence to claim that YAI4Hu is better than the XAI-based explainer (U=40, p=.0002) on those questions. The same can be said about the two-level explainer (U=48, p=.003) and the how-why explainer (U=65.5, p=.02).

The difference between a normal XAI-based explainer and the other explainers is twofold. First, the explanations produced by YAI4Hu and the how-why explainer are interactive and more user-centred, while those of the normal XAI-based system are not. Secondly, the normal XAI-based explainer considers a smaller amount of explainable information. YAI4Hu, the how-why explainer and the two-level explainer produce their explanations using more than 50 extra web pages that the XAI-based explainer does

not see. In particular, the amount of information the normal XAI-based explainer handles is $\frac{1}{100}$ of all the other explainers. This last difference allows us to exploit these user studies further to test Hypothesis 2 (cf. Chapter 4).

In order to show that an increment in DoX causes a consequent increment in the effectiveness of explanations, we have to compute the DoX scores of the normal XAI-based explainer and the DoX scores of the other explainers involved in the user study. To do so, we identified the set of explanandum aspects A from the quizzes used to generate the effectiveness scores (see Tables 7.1 and 7.2). These quizzes define what the users should know to be effective, indirectly defining what is essential for the system to explain: the explanandum aspects.

Eventually, if Hypothesis 2 holds, we would expect that the greater DoX is, the greater the effectiveness of an explainer. Notably, the opposite is not necessarily correct. Two explainers (with different presentation logics; e.g., the two-layered explainer and YAI4Hu) might have different effectiveness scores despite having the same DoX.

Computing the DoX scores for this second experiment, we got the results shown in Table 8.2. Importantly, these results confirmed our expectations for them. They indicate that the two-level explainer, the overview-based explainer, the how-why explainer and YAI4Hu have higher DoX scores than the normal XAI-based explainer regardless of their presentation logic.

8.4 Discussion and Analysis of Empirical Results: How to Use DoX for Assessing Law Compliance

The results of all experiments and user studies showed that Hypothesis 2 (cf. Chapter 4) is valid. We see that DoX increases whenever a black-box AI is enclosed in a XAI and that an increase in DoX corresponds to a statistically significant increase in the effectiveness of the explanatory system. Therefore, our technology for estimating the DoX might be used for an objective and lawful algorithmic explainability assessment as soon as what is needed to be explained can be identified under the requirements of the law in the form of a set of precise *explanandum aspects*. To guarantee the reproducibility of the experiments, we published the source code of DoXpy[2], as well as the code of the XAI-based systems, the user study questionnaires

[2] `https://github.com/Francesco-Sovrano/DoXpy`

Table 8.2: ***Results of the 2nd experiment on DoXpy.*** *The scores in this table are different from those of the first experiment (Table 8.1) because a different explanandum is considered for the second experiment. In this table, DoX and average (Avg) DoX are shown for the credit approval system (CA) and the heart disease predictor (HD). As columns, we have the normal XAI-based explainers (XAI, for short) and the other explainers, i.e., YAI4Hu, the two-level explainer and the how-why narrator. For more details about interpreting this table, read the caption of Table 8.1.*

		CA		HD	
		XAI	Others	XAI	Others
Avg DoX	ML	0.5	7.45	0.17	8.65
	TF	0.22	6.41	0.09	9.03
DoX	ML	"how": 0.53 "which": 0.53 "what": 0.52 "why": 0.51 "when": 0.49 "who": 0.48 "where": 0.46 "whose": 0.46	"how": 7.94 "what": 7.89 "which": 7.76 "why": 7.64 "when": 7.63 "whose": 7.06 "where": 6.88 "who": 6.76	"which": 0.19 "whose": 0.18 "how": 0.18 "what": 0.17 "why": 0.17 "when": 0.17 "where": 0.17 "who": 0.17	"why": 9.48 "which": 9.31 "how": 9.11 "what": 8.72 "whose": 8.55 "when": 8.45 "where": 8.02 "who": 7.93
	TF	"what": 0.28 "when": 0.25 "how": 0.20 "who": 0.19 "where": 0.17 "which": 0.17 "whose": 0.13 "why": 0.11	"what": 8.41 "when": 7.10 "how": 5.77 "who": 5.75 "where": 5.23 "which": 5.01 "why": 3.97 "whose": 3.81	"what": 0.11 "when": 0.09 "how": 0.09 "who": 0.09 "where": 0.09 "why": 0.08 "which": 0.08 "whose": 0.07	"what": 11.00 "when": 9.17 "how": 8.34 "who": 8.09 "which": 8.01 "where": 7.64 "why": 7.44 "whose": 6.17

and the remaining data mentioned within this chapter.

The results of the first experiment tell us that whenever new information about different aspects to be explained is added to the explanandum support material, the DoX scores increase, and this is also true when changing the set of explanandum aspects, as we did with the second experiment. Furthermore, the results of the second experiment tell us that whenever the DoX scores increase, the overall effectiveness of the explanations generated from the explanandum support material increases as well. This is true even for the two-level explainer, even though it is not interactive and does not reorganize information to make it simpler and easier to access, dumping on the user dozens of pages of content.

Our user studies involved more than 190 participants and were consistent across two somewhat different and broad user pools, producing statistically significant results (with *p*-values lower than .05). Therefore, considering that *explainability* is fundamentally the *ability to explain*, the two experiments combined tell us that our (average) DoX can quantitatively approximate the degree of explainability of information. In other words, we conclude from our experiments that DoX *can* be used as a proxy for measuring the explainability of an explanatory system, as long as a set of explanandum aspects can be defined. Moreover, DoX is deterministic and fully objective, and it could be used as a cheaper alternative to expensive non-deterministic user studies.

We are convinced that DoX may have a role in all applications where it is essential to evaluate explainability objectively. Indeed, the main benefit of DoX is that it works with any set of explanandum aspects A. Therefore it can be used to quantify how the explanations given by an AI system are aligned with any of the Business-to-Business and Business-to-Consumer requirements identified by Bibal et al. [22].

For each **Business-to-Business and Business-to-Consumer** requirement we may have the following set of **explanandum aspects** A:

- **Providing the main features used in a decision by the AI**: A can be the set of main feature labels used for a decision. This list can be generated with a XAI like CEM, TreeSHAP or others.
- **Providing all features processed by the AI**: in this case, A is the set of all the feature labels considered by the AI.
- **Providing a comprehensive explanation of a specific decision taken by the AI**: A can be the set of aspects deemed relevant to the decision of the AI, i.e., what is the AI, what are the known issues of the AI, or all the other aspects discussed in Chapter 1.
- **Providing the underlying logical model followed by the AI**: in this case, A can be the set of all the nouns or noun/verbal phrases used in the textual description of the logical model of the AI.

Hence, the benefits of using DoX over a normal user study are manifold, in fact:

- DoX reduces testing costs normally sustained during subject-based evaluations.

- DoX allows the direct measurement of the degree of explainability of any piece of information for which a meaningful textual representation is written in a natural language (i.e., English).
- DoX disentangles the evaluation of the explanandum support material from that of the explainer (or presentation logic) and the interface.

In other words, DoX is a fully objective metric that could be used to understand whether a piece of information is sufficient to explain something regardless of whether the resulting explanations have been perceived as satisfactory and good by the explainees. We deem this characteristic of DoX to be very important. A poor degree of explainability objectively implies poor explanations, no matter how good the adopted explanatory process is (or how it is perceived): "Users also do not necessarily perform better with systems they prefer and trust more. To draw correct conclusions from empirical studies, explainable AI researchers should be wary of evaluation pitfalls, such as proxy tasks and subjective measures" [37].

The results of the second experiment show that explanatory systems with the same DoX could be usable and effective in different ways. This indicates that DoX should not be considered as a total replacement to user studies but rather as a cheaper alternative to consider while developing complex explanatory systems. In other words, DoX cannot fully replace subjective metrics (i.e., usability) if one wants to evaluate the user-centrality of an explanatory system or interface. Instead, DoX is probably better than subjective metrics if one wants to objectively evaluate the contents of an explanatory system to understand how many questions can be adequately answered. The higher DoX, the greater the chances to sufficiently explain to various users.

Furthermore, we emphasize the benefits of using DoX as an early testing metric for designs. If DoX is low for a particular approach, it is highly unlikely to score well on usability (effectiveness) later on. Given the relative ease with which DoX can be measured compared to usability studies, this advantage should not be underestimated.

Despite the considerable benefits of DoX, which are supported by both theoretical and empirical evidence, we must also acknowledge its limitations. One such limitation is the challenge of accurately defining the explanandum aspects. We must also consider the potential sensitivity of our algorithm for calculating DoX scores to the selection of a deep language model for relevance estimation. This is suggested by the numerical discrepancies between the DoX scores in Tables 8.1 and 8.2.

On the one hand, we see that the difference in terms of DoX between the normal XAI-based explainers and the other explainer tend to differ from MiniLM to the Universal Sentence Encoder slightly. On the other hand, we also see that in all the considered experiments, the DoX scores increase as expected, with both MiniLM and the Universal Sentence Encoder, suggesting that the alignment of DoX to explainability is independent of the chosen deep language model. This intuition is supported by the fact that the deep language models, on average, perform reasonably well on existing benchmarks for evaluating answer retrieval algorithms. In other words, if the average DoX aggregates enough archetypes, aspects and details, then different pertinence functions performing similarly on standard benchmarks may produce proportionally similar scores. This does not exclude the fact that some deep language models might be better than others for computing DoX scores or that multiple standardized deep language models should be adopted for a thorough estimate of the DoX. We leave this analysis for future work.

Another possible limitation of DoX is that its scores cannot be easily normalised in a $[0, 1]$ range. In fact, according to Definition 6 (cf. Chapter 4), DoX is computed by performing a sum (called *cumulative pertinence*) over the set of details D extracted from an explanandum support material. So, DoX can measure the similarity of the explanandum support material to the explanandum. Unfortunately, it is not possible to know in advance the total number of details of any possible explanandum support material. Therefore, it is impossible to normalize the score by dividing the cumulative pertinence by such a number. It is worth noting that such a sum is necessary. Indeed, suppose the cumulative pertinence was a mean instead of a sum. In that case, the resulting score for an explanandum support material could not be compared to that of any larger (in terms of the number of details) explanandum support material, making pointless the use of DoX in the first place.

Furthermore, it is essential to mention that DoX, alone, is not sufficient for a thorough quantification of how much of the information is explained by an AI system. Our definition of DoX does not consider the correctness of information of the explanandum support material, assuming that truth is given and that it is different from explainability. In other words, DoX should always be used with other metrics that can evaluate the correctness of available information.

Finally, although DoX can be used to verify many of the requirements

defined by Bibal et al. [22], it is still unclear how to apply DoX to verify also Government-to-Citizen legal requirements. Additionally, selecting a reasonable threshold of DoX scores for law compliance is undoubtedly one of the following challenges we envisage for a proper *standardization of explainability* in the industrial context. We also leave these analyses for future work.

CHAPTER 9

Identification and Evaluation of Strategies for Retrieving Answers from Technical Documents

In the previous chapters, we have seen how to use question-answering algorithms based on neural reading comprehension to explain different topics (e.g., finance and healthcare). To do so, we relied on deep language models[1] pre-trained on a large variety of (mostly) non-technical textual resources: *i)* the Stanford Natural Language Inference corpus [28], *ii)* the

The work presented in Chapter 9 was developed in collaboration with Salvatore Sapienza, and Vittoria Pistone from the University of Bologna [200]. *F. Sovrano*: conceptualization, methodology, software, original draft preparation, visualization, investigation, validation, review and editing. *S. Sapienza*: the methodology used for the creation of the Q4Eu dataset. *S. Sapienza*: the Q4eIDAS and Q4GDPR datasets. *V. Pistone*: the Q4EAW dataset and assistance with the error analysis of DiscoLQA. The Q4PIL comes from [197], created by *B. Distefano* from the University of Bologna and *S. Sapienza*.

[1] A (deep) language model is a deep neural network trained in an unsupervised manner to capture and represent a language domain, learning how words are statistically distributed in collections of documents.

Natural Questions corpus [113], *iii)* TriviaQA [103], *iv)* the WebQuestions corpus [19] and *v)* CuratedTREC [16].

Neural reading comprehension models and (more generally) data-centred machine learning can learn from raw data, with performance improving in proportion to the quantity and quality of the data acquired. When not enough data is available or completing an entire training procedure is too expensive in terms of computational effort, a commonplace in natural language processing is to fine-tune (deep) language models. These models are usually pre-trained on generic non-technical documents and then trained in a supervised manner on downstream tasks [93].

Nevertheless, what if only a small dataset is available that is insufficient to train or retrain a pre-trained language model? This situation is not uncommon when legal English (i.e., *legalese*) or other specialised variants of natural languages are involved in tasks requiring automated processing or understanding. Neural reading comprehension of legal or other technical texts is challenging because legalese and technical languages are rarer, mercurial and in many ways different from commonly used natural languages.

Specialised language variants share many similarities with their corresponding base languages. Thus, fine-tuning a general-purpose pre-trained model can undoubtedly aid in handling those aspects of the technical language that are similar to its ordinary language. However, it is hard to believe that fine-tuning general-purpose language models with small datasets would suffice, or it would be even beneficial, to train specialised models capable of generalising on unseen data [46].

Indeed, the difference between technical and ordinary languages fosters issues when applying or fine-tuning general-purpose language models, i.e., for open-domain question-answering. This is especially true when the meaning of a technical document (e.g., a textbook or a law) is encoded in its (discourse) structure in a way that is different from the spoken language, e.g., the one used daily in social media, forums and blogs. For example, long sentences or more "formal" writing may be preferred in legal English (e.g., Brussels I bis Regulation EU 1215/2012) to reduce potential ambiguities and improve comprehensibility. Nevertheless, the noise introduced by the excessive length of the sentence or their unusual structure can distract a language model trained in ordinary English, pushing it to commit more errors.

Consequently, in these language variants, it is common to find out that the minimal training set that needs to be annotated manually for adequate

deep-learning tasks ends up being the same size as the whole corpus. An example of this could be the corpus of the United Nations General Assembly Resolutions, comprising only a few thousand resolutions written over several years by different authors and with various language constructs and vocabulary choices.

Therefore, we hereby investigate some mechanisms to perform "zero-shot" question-answering on technical documents to apply it effectively to our YAI and case studies. We do so by focusing on legalese because it is the main subject of the case study presented in Chapter 10. It is also the technical language used by the GDPR, the AI Act and other legislative texts to which many YAI should be compliant (cf. Chapter 1).

Specifically, "zero-shot" means that question-answering is performed through pre-trained language models without fine-tuning them on the downstream (technical) task of question-answering. In this sense, zero-shot question-answering can be a necessary solution for all those tasks characterised by a paucity of data (e.g., European hard laws, the resolutions of the United Nations General Assembly) and for which we want to train AI-based solutions through machine learning without having enough information for effective fine-tuning. Conversely, zero-shot question-answering might be less useful whenever data are abundant (e.g., American case law or privacy policies).

Zero-shot question-answering is an alternative approach to few-shot question-answering, where few (in the order of 10) examples are used to fine-tune a language model. Although few-shot learning may be a good compromise solution to the data scarcity problem, as pointed out by Chowdhery et al. [51] or Wang et al. [221], for now, it seems to be a viable solution only with huge language models, such as PaLM[2] [51], pre-trained on thousands of billions of tokens of high-quality text. In practice, this technology can only be used with highly specialised hardware and sophisticated computing capabilities like those of companies such as Google, Microsoft, and Meta.

Conversely, we address the problem of data scarcity in processing and understanding texts written using various technical legalese constructs by starting from the following hypothesis.

Hypothesis 4. *Technical language (i.e., legalese) is similar to its base language, and its meaning does not deviate much from the spoken language,*

[2]The number of parameters learned by PaLM is 540 billion.

except for certain constructions of words. In particular, legalese constructs play with syntagmatic relations[3] *in a unique way. This fact can be exploited to tackle the data scarcity problem.*

In addition, we delve into the role of discourse structure in technical languages, particularly legalese, seeking to understand and exploit its importance in encoding the meaning of technical documents. Specifically, we study what happens when changing the type of information to consider for answer retrieval during question-answering. Therefore we also make the following hypothesis.

Hypothesis 5. *Suppose a language model is not specialised in legalese or other technical languages. In that case, it may likely fail to identify and capture the importance of specific grammatical sub-trees (i.e., clauses) that are not common to a spoken language. Hence, by selecting those grammatical sub-trees deemed the most important, we should be able to help the information retriever and question-answering system by partially hiding noise within answers. To identify these grammatical sub-trees, we can use theories of discourse [157] and sentential meaning representation [15].*

In the following sections, we will present new technological solutions, based on Hypotheses 4 and 5, for more affordable YAI tools based on answer retrieval. To evaluate them, we considered English legalese as the technical language for the case study, using a new dataset called Questions for European Legislation (Q4EU for short).

9.1 Q4EU: a Dataset for Evaluating Answer Retrieval on European Legislation

Q4EU is a dataset for evaluating answer retrieval algorithms. It comprises 72 unique questions[4] and 225 expected answers (i.e., articles and recitals) on 6 heterogeneous European norms spanning from Private International Law to Human Rights Law (i.e., the General Data Protection Regulation, UE 2016/679), from regulations of electronic signatures to the European arrest warrant. For simplicity of exposition, **Q4EU can be divided into the following datasets**.

[3] Syntagmatic associations indicate compatible combinations of words (i.e., the word "rotten" combined with "apple"), excluding others (i.e., the syntagm "curdled apple").

[4] The minimum number of queries required for a good information retrieval test set, in order to obtain statistically significant results, usually is 50 [52].

Q4PIL (see Table 9.5): containing questions about 3 private international laws: Rome I Regulation EC 593/2008; Rome II Regulation EC 864/2007; Brussels I bis Regulation EU 1215/2012. These regulations are, respectively, on the law applicable to contractual obligations, on the law applicable to non-contractual obligations, on jurisdiction and the recognition and enforcement of judgements in civil and commercial matters. In particular, they aim to provide a tool for identifying the applicable law and the jurisdiction in cases when two or more legal systems connect and generate complex relationships (e.g., a sale of goods contract between an Italian and a German citizen regarding commodities situated in Spain).

Q4EAW (see Table 9.3): containing questions about the Council Framework Decision of 13 June 2002 on the European arrest warrant and the surrender procedures between Member States[5]. This framework decision increases the efficiency of extradition procedures for crime suspects. Furthermore, it also determines the abolition of formal extradition procedures between member states of the EU for persons who are fugitives from justice after being finally convicted. The framework decision represents the first concretisation of the principle of free movement of judicial decisions in criminal matters, encompassing both pre-judgement and final decisions by fostering judicial cooperation and the development of a single area of freedom, security and justice in the EU.

Q4GDPR (see Table 9.4): containing questions about the GDPR (cf. Section 1.1), the most relevant piece of legislation in the EU legal framework with regards to data protection law. Its goal is to foster the fundamental right to data protection, enshrined by the Charter of Fundamental Rights of the European Union (Art. 8), while harmonising rules in data processing, profiling, and risk management.

Q4eIDAS (see Table 9.2): containing questions about Regulation (EU) No 910/2014 of the European Parliament and of the Council of 23 July 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing Directive 1999/93/EC,

[5] https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:02002F0584-20090328&from=EN

also known as eIDAS Regulation[6]. This legislation tackles several issues in electronic identification, electronic signature, electronic seals, and trust services. Its goal is to provide legal certainty for cross-border transactions in the EU Single Market.

Some statistics on the sets mentioned above are shown in Table 9.1.

Table 9.1: ***Statistics on Q4EU.*** *The column "Art./Rec." counts the number of recitals and articles. The column "Questions" counts the number of different questions, and the column "Tokens per Art./Rec." counts the mean number of tokens per article/recital, and so on. Please note that Q4EU is the sum of Q4PIL, Q4EAW, Q4GDPR and Q4eIDAS.*

	Questions	Expected Answers	Answers per Question	Norms	Art./Rec.	Tokens	Tokens per Art./Rec.
Q4PIL	17: 5 low; 7 normal; 5 high	65	3.82	3	269	27,280	101.41
+ Q4EAW	21: 7 low; 7 normal; 7 high	68	3.23	1	50	8,426	168.52
+ Q4GDPR	17: 4 low; 7 normal; 6 high	55	3.23	1	272	45,138	165.94
+ Q4eIDAS	17: 5 low; 7 normal; 5 high	37	2.17	1	129	17,283	133.97
= **Q4EU**	72: 21 low; 28 normal; 23 high	225	3.12	6	720	98,127	136.28

To build the Q4EU dataset, the pieces of legislation (i.e., the norms) kept into account are conceived as self-contained legal environments. While legal interpretation is often grounded on external legal factors (e.g., jurisprudence, scholars' opinions), we opted for a "black letter" approach to the law that only considers the legislative legal formant. Therefore, the point of view assumed in our analysis is the perspective of the lawmakers. This has a twofold implication for question-and-answer drafting.

Questions have been modelled to be answered solely within the legal text under scrutiny. They do not refer to legal concepts that are not explicitly mentioned in the regulations, such as the hierarchy of legal sources or competence. Moreover, not all the (legal) questions are the same. While some accept as an answer a provision that exactly matches the question, others rely on more complex interpretations (i.e., legal reasoning) to be

[6] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32014R0910

answered. Therefore, questions have been classified depending on their context specificity, which can either be low, normal, or high.

First, specific questions whose answer is precisely in the domain of the regulations were labelled as *highly* specific. An example of a question with high specificity is "In what court can an employee sue its employer?" because it perfectly falls within the scope and goals of Regulation Brussels I-bis and finds its exact answer in the provisions of Articles 21 and 23.

Questions whose answer falls within the scope of the regulations while requiring an abstraction of multiple legal provisions were labelled as *normally* specific. For instance, "What is the applicable rule to protect the weaker party of a contract?" was labelled as normally specific. This is because its answer also relies on the concept of "weaker party" mentioned across two regulations (Recital 23 Rome I and Recital 18 Brussels I) concerning any contract (as a legal concept) rather than specific contractual types.

Finally, broad questions whose tentative answer is found through an articulate combination of articles and recitals were labelled as having *low* specificity. For instance, a question with low specificity is "Can the parties choose a different applicable law for different parts of the contract?". While Rome I Regulation provides for a discipline on the applicable law to contract, it does not contain any provision concerning individual parts. The answer is ultimately open to interpretation in such a question, whereas the Regulation suggests norms that could serve as a reference point.

Since such classification might be subjective and dependent on each jurist, three legal experts independently evaluated the level of context specificity and decided by the majority about the final level.

Instead, the answers to the questions provided by legal experts are obtained by mirroring the question-drafting methodology. Three legal experts, different from the question-drafters, provided answers to the legal questions by looking for the following: *i)* specific, punctual, and explicit answers in the case of high-specific questions; *ii)* general and conceptual, yet text-based, answers to normally specific questions; *iii)* prima facie textual references to be used as interpretative points of reference in the case of low specific questions.

These experts only provided textual references in the legislation at the article or recital level (e.g., Rome I Art. 8; B Rec. 18). When at least two experts agreed on a given answer, their response was considered valid without further enquiry. If one expert provided a different answer, another

expert validated this response. In drafting the validation answers, no other articles or recitals were considered except those provided by the original validators.

Table 9.2: ***Q4eIDAS Subset.*** *Here, "E" is the eIDAS Regulation. "Rec." means Recital, while "Art." means Article. The column "S." indicates the specificity of the questions: "L" means low specificity, "N" means normal and "H" means high. The column "T." indicates which norm (i.e., regulation or decision) a question targets. If no norm is indicated next to the article/recital, then the norm of the article/recital is indicated in the "T." column.*

Question	S.	Expected Answers	T.
How is a qualified electronic signature validated?	H	Art.32, Art.33, Rec.57	E
Can an electronic signature be expressed in the form of a pseudonym?	N	Art.3.14, Art.32	E
Can a minor obtain a qualified electronic signature?	L	Art.3, Art.25	E
From when qualified certificates lose their validity in the case of revocation?	N	Art.24, Art.28	E
Is a graphometric signature qualified as an advanced electronic signature?	L	Art.3.11, Art.26	E
How should access to trust services be granted to persons with disabilities?	N	Rec.29, Art.15	E
How can the identity of a natural person be verified in the issuing of a qualified certificate?	H	Art.24.1	E
Do electronic contracts have the same validity as paper contracts?	L	Rec.21, Art.2.3	E
Why is there a specific discipline for the notification of security breaches?	H	Rec.38, Art.19.2	E
When shall a trust service provider notify affected individuals and users?	H	Art.19.2	E
What is the applicable law to the trust service provider which provides its trusted services in a Member State different from the one where it is established?	L	Rec.22, Rec.42, Art.4, Art.6, Art.24	E
How can qualified certificates be temporally limited?	N	Rec.53, Art.24.4, Art.28, Art.38.5	E
What are the requirements for website authentication?	N	Rec.67, Art.45	E
When do electronic signatures qualify as "advanced electronic signatures"?	N	Art.3.11, Art.26	E
Which subject has the competence to maintain trusted lists?	H	Art.22	E
How should liability be determined for Member States that are non-compliant with provisions about electronic identification schemes?	N	Rec.18, Art.11	E
What is a security breach?	L	Art.10, Art.19	E

Table 9.3: ***Q4EAW Subset.*** *Here, "W" is the Council Framework Decision on the European arrest warrant. For more details on how to read this table, please see the caption of Table 9.2.*

Question	S.	Expected Answers	T.
What is the European arrest warrant?	N	Art.1.1, Art.8, Art.9.3, Rec.11, Rec.6	W
Can the execution of the European arrest warrant be refused when the law of the executing Member State does not impose the same type of tax or duty or does not contain the same type of tax rules as the law of the issuing Member State?	L	Art.2.2, Art.2.4, Art.4.1, Rec.6	W
Who decides precedence in the event of a conflict between a European arrest warrant and a request for extradition from a third country?	N	Art.16.3, Rec.8, Art.10.6	W
Which law is used to record the consent to surrender of a requested person?	H	Art.13.3, Art.11	W
Is the arrest warrant based on the principle of mutual recognition?	L	Rec.2, Rec.6, Rec.5, Art.1.1, Art.1.2, Rec.10	W
Does a requested person have the right to an interpreter?	H	Art.11.2	W
Can the consent to the surrender of the arrested person be revoked?	N	Art.13.4, Art.17	W
Is the surrender of the arrested person always subject to the verification of the double criminality of the act?	L	Art.2.2, Art.2.3, Art.2.4, Art.4.1, Art.5, Art.33	W
Which authority should be informed in case of repeated delays by a Member State in executing European arrest warrants?	H	Art.17.7	W
Can the Member States also apply other agreements in addition to the Framework Decision?	L	Art.31, Rec.5, Art.33, Art.32	W
Can the European arrest warrant be ordered for the execution of a non-custodial sentence?	N	Art.2.1, Art.1.1, Rec.12, Art.5	W
Can the executing judicial authority refuse to execute the European arrest warrant when the person who is the subject of the European arrest warrant is being prosecuted in the executing Member State for the same act as that on which the European arrest warrant is based?	N	Art.4.2, Rec.8, Art.24, Rec.13	W
What right is applied by the judicial authority to decide whether the requested person should remain in detention or be provisionally released?	H	Art.12.1, Rec.8, Rec.10	W
Can the constitutional rules of the Member States be applied?	L	Rec.7, Rec.12, Art.1.3, Art.34	W
Should the European arrest warrant be translated into the official language or one of the official languages of the executing Member State?	H	Art.8.2, Rec.8	W
Can the executing judicial authority request the opinion of Eurojust in case of multiple requests?	H	Art.16.2, Rec.8	W
Can the executing judicial authority, on its own initiative, seize and hand over property acquired by the requested person as a result of an offence?	N	Art.29.1, Rec.5	W
Is an alert in the Schengen Information System equivalent to a European arrest warrant?	N	Art.9.3, Art.8.1, Art.1.1	W
What are the time limits for the surrender of the requested person?	L	Art.23, Art.15, Art.17, Art.20, Art.24, Rec.1	W
How are the expenses of executing the European arrest warrant allocated?	H	Art.30	W
What claims can be made to the judicial authority by the interested party who has not previously received any official information on the existence of the criminal proceedings against him/her?	L	Art.4a, Rec.12, Art.11	W

Table 9.4: ***Q4GDPR Subset.*** *Here, "G" is the GDPR. For more details on how to read this table, please see the caption of Table 9.2.*

Question	S.	Expected Answers	T.
Does the GDPR provide a right to explanation?	L	Rec.71, Art.12.3, Art.15.1	G
When is it mandatory to carry out a Data Protection Impact Assessment?	H	Art.35.1, Art.35.3	G
What are the possible security measures that can be adopted to mitigate the risks related to personal data processing?	N	Art.32.1, Art.32.2	G
What are the applicable rules to the processing of personal data for archiving purposes in the public interest, for scientific or historical research purposes or for statistical purposes?	L	Rec.156, Art.5.1.b, Art.9.2.j, Art.14.5.b, Art.17.3.d, Art.89	G
How should a data processor be appointed?	N	Art.26, Art.38	G
When is the consent of the data subject explicit?	L	Rec.51, Rec.71, Rec.111, Art.7.1, Art.9	G
What elements shall the European Commission keep into account to authorise the transfer of personal data to a third country through an Adequacy Decision?	N	Art.45.2, Art.45.3, Rec.104	G
What are the rules applicable to biometric data?	H	Rec.51, Rec.53, Art.9	G
When does the public interest override data subject rights?	L	Rec.45, Rec.46, Rec.50, Rec.65, Rec.69, Art.9.2.i, Art.17.3, Art.89	G
To what data is the right to portability applicable?	H	Art.20	G
How should a data processing record be drafted?	H	Art.30	G
What data processing poses significant risks to the fundamental rights and freedoms of natural persons?	N	Rec.51, Rec.75, Art.9, Art.10	G
What elements should be included in a Code of Conduct?	N	Rec.81, Art.40	G
What are the obligations of the data controller when the legal basis for the data processing is the consent of the data subject?	N	Art.7, Art.13, Art.14, Art.20	G
Which legal entity can impose fines on data controllers?	H	Rec.130, Art.58.2.i, Art.83	G
Who can exercise the right to lodge a complaint before the supervisory authority?	N	Rec.141, Rec.142, Art.77	G
What is the procedure to follow in the event of a data breach?	L	Rec.85, Rec.86, Rec.87, Rec.88, Art.33, Art.34	G

Table 9.5: ***Q4PIL Subset.*** *Here, "B" is Brussels I, "RI" is Rome I, and "RII" is Rome II. For more details on how to read this table, please see the caption of Table 9.2.*

Question	S.	Expected Answers	T.
Who determines disputes under a contract?	L	Art.7.1, Art.8.3, Art.8.4, Art.17	B
What factors should be taken into account for conferring the jurisdiction to determine disputes under a contract?	N	Art.7.1, Art.17, Art.20, Art.25	B
Which parties of a contract should be protected by conflict-of-law rules?	N	Rec.23, Art.6, Art.8, Art.13	RI
In which case are claims so closely connected that it would be better to treat them together in order to avoid irreconcilable judgments?	H	Art.8, Art.30, Art.34	B
What kind of agreement between parties is regulated by these Regulations?	L	B Rec.6, B Rec.10, B Rec.12, B Art.1, RI Rec.7, RI Art.1	B, RI, RII
In which court is celebrated the trial in case the employer is domiciled in a Member State?	H	Art.21, Art.22, Art.23	B
How should a contract be interpreted according to Regulation Rome I?	L	Rec.22, Rec.12, Rec.26, Rec.29, Art.12	RI
Which law is applicable to a non-contractual obligation?	N	Rec.17, Rec.18, Rec.26, Rec.27, Rec.31, Art.4-20	RII
Can the parties choose the applicable law in consumer contracts?	H	Rec.11, Rec.25, Rec.27, Art.6	RI
What factors should be taken into account for conferring the jurisdiction to determine disputes under a consumer contract?	N	Rec.18, Art.17, Art.18, Art.19, Art.26	B
Can the parties choose a different applicable law for different parts of the contract?	L	Rec.11, Art.3.1	RI
What non-contractual obligations fall into the scope of Regulation Rome II?	H	Rec.10, Rec.11, Art.1, Art.2	RII
What is the applicable rule to protect the weaker party of a contract?	N	RI Rec.23, B Rec.18	B, RI
What is the applicable law to determine the validity of consent?	L	Art.3.5, Art.10, Art.11, Art.13	RI
When are two actions to be considered related according to the Regulation Brussels I Bis?	N	Rec.21, Art.30.3	B
What court has jurisdiction in case of a counter-claim?	N	Art.8.3, Art.14.2, Art.18.3, Art.22.2	B
Where can an employee sue their employer?	H	Rec.14, Rec.18, Art.21.1, Art.22.1, Art.23	B

9.2 SyntagmTuner: Combining Shallow and Deep Learning Approaches against Data Scarcity

State-of-the-art natural language processing is often based on statistical semantics, which has its roots in what is called *distributional hypothesis*. The distributional hypothesis is a rather important concept according to which the meaning of words depends statistically on their context and co-occurrences [86]. The distributional hypothesis can be rephrased as follows: "a word is characterised by the company it keeps" [72].

The use of statistical semantics over text documents has led to the discovery of technologies capable of encoding the meaning of words and documents as numerical representations. These numerical representations are called *embeddings*, i.e., mathematical objects that can be represented as multi-dimensional points in a mathematical (e.g., Euclidean) space so that classical mathematical operations can be operated on them. For example, by computing the distance (or the cosine similarity) between two of these embeddings (i.e., points in a mathematical space), it is possible to quantitatively estimate the degree of similarity between the meaning of their corresponding words or document fragments.

The distributional hypothesis is one of the fundamental gears behind the performance of deep language models. Its impressive compatibility with deep learning technologies is why the distributional hypothesis, originating in computational linguistics, is now receiving attention also in Cognitive Science [133].

Several techniques exist for learning numerical representations of texts from their occurrence information. Some specialise in words, others in longer text fragments such as sentences or entire documents. Existing models could be broadly grouped into two categories [174]. The first category leverages more on the *syntagmatic relations* between words, which relate to words that co-occur within the same text region [205]. In contrast, the second one leverages more on the *paradigmatic relations*, which relate to words that occur within similar contexts but may not co-occur anywhere in the text.

One of the oldest and most basic techniques for sentence embedding is probably *Bag of Words* (BoW for short) [86]. A BoW is a non-ordered set of words representing individual occurrences, disregarding grammar and even word order but paying attention only to frequency. An example of Bag of Words embedding for the sentence "This sentence is cool even if a

sentence" could be:

```
this = 1;
sentence = 2;
is = 1;
and = 0;
gibberish = 0;
cool = 1;
even = 1;
if = 1;
a = 1.
```

As we can see in the example, Bag of Words tokenises documents, counting the number of token occurrences and then returning them as a (sparse) numerical matrix.

Another technique for sentence embeddings is *Term Frequency–Inverse Document Frequency (TF-IDF)* [102]. Specifically, TF-IDF is computed as the product of two statistics: *Term Frequency* (TF) and *Inverse Document Frequency* (IDF). Term Frequency is the output of a Bag of Words model. For a specific document, TF estimates how important a word is by looking at how frequently it appears. The Inverse Document Frequency, on the other hand, is based on the idea that essential words for a specific document (also called signature words) frequently appear inside this document but likely less often inside other documents. The frequency of signature words is usually low across different documents; thus, its Inverse Document Frequency must be high. Therefore, the similarity between TF-IDF embeddings is said to be *syntagmatic* [174, 205] since it concerns words that co-occur within the same text region (e.g., the same sentence, paragraph, or document).

One of the main issues with (vanilla) TF-IDF is that it is biased against long documents [186], tending to favour retrieval of short documents and suppressing the retrieval of long documents. Nonetheless, techniques exist such as *pivoted document length normalisation* [186] to counteract this unwanted bias by intelligently giving a smaller weight to shorter documents and a larger weight to longer documents.

Another issue with TF-IDF and Bag of Words is that they usually generate very sparse embedding matrices, depending on the size of the context snippets [155], which are difficult to handle efficiently on large-scale datasets. To this end, techniques such as *Latent Semantic Analysis* (LSA) [68] can reduce the sparsity of embeddings. Nonetheless, TF-IDF, as well as LSA, perform poorly on capturing the meaning of words encoded in

paradigmatic relations, also not being sensitive to word order.

More paradigmatic approaches for document embedding capable of considering word order are the encoder-decoder models based on neural networks, e.g., *paragraph vectors* [117] or *skip-thought* vectors [110]. Recently, with the realisation that deeper neural networks can perform better embeddings, researchers have started to devise more complex and performant sentence/document embedding techniques, such as the Transformers [211], or other derivative works, e.g., BERT [60], Universal Sentence Encoder [225, 44], T5 [161].

To summarise, on the one hand, there are shallow syntagmatic techniques for text embedding, such as TF-IDF, that are easy to train in an unsupervised manner. On the other hand, there are also deeper and paradigmatic techniques, such as BERT or the Universal Sentence Encoder. These are based on deep learning techniques, which can be as effective as they are complex and expensive, requiring, in many cases, large amounts of training data, specialised hardware and many hours (e.g., weeks) of learning to achieve the best performance.

However, if Hypothesis 4 is correct, we could improve the performance of pre-trained general-purpose deep language models by simply combining them with ad hoc models for capturing the patterns of syntagmatic relations across texts. It would be possible without the need to re-train any deep language model. The point is that such syntagmatic relations can be identified even with little data, e.g., by shallow machine learning techniques such as TF-IDF. These simple tools can be quickly trained in an unsupervised manner on the available data. They can capture a part of the meaning that, in legalese or other technical languages, is encoded into syntagmatic information. On the contrary, more sophisticated tools (e.g., deep neural networks) (pre-)trained on generic natural language can capture parts of meaning that are not peculiar to the technical variant of the base language.

In particular, we could use TF-IDF to model domain-specific information in combination with a Universal Sentence Encoder (or any other state-of-the-art deep language model) to model generic information (e.g., semantic relations between non-domain-specific words). Interestingly, combining TF-IDF with deep language models is not entirely new. For example, Kowsari et al. [112] and Du et al. [67] proposed to exploit TF-IDF for improving the training of an artificial neural network. While others [229, 117, 10] proposed to use TF-IDF for computing *weighted word embeddings* (i.e., a type of averaged word embedding).

In contrast, we propose to combine similarities instead of embeddings and to exploit pre-trained paradigmatic models without resorting to retraining or fine-tuning procedures for downstream tasks. We call this technique *SyntagmTuner*. Specifically, in SyntagmTuner, the similarity between snippets of text is computed as a (linear) combination of: the embedding obtained with pre-trained general-purpose language models, and the embedding of a shallower model (i.e., TF-IDF with pivoted document length normalisation) capable of generating more syntagmatic similarities. In this way, the TF-IDF similarity plays out as a kind of *topic-related* similarity extracted by populating the embeddings with information about the "regions of the text in which the linguistic elements are found". On the contrary, the similarity between the embeddings of a deep language model plays out as a kind of *paradigmatic* similarity extracted by populating the embeddings with information about "which other linguistic elements the items co-occur with". In other words, the idea behind SyntagmTuner is to combine the unique and different properties of the similarities mentioned above to obtain a new paradigmatic similarity potentially expressing a topic-related similarity in a domain on which the deep language model has not been trained.

SyntagmTuner is a pipeline of AI techniques, consisting of two main phases, as shown in Figure 9.1. The syntagmatic model (i.e., TF-IDF) is built during the first phase, together with the knowledge graph of the answer retriever. This phase consists of unsupervised training of the TF-IDF model on a corpus of documents in order to identify the *signature words*[7] contained within it. Notably, the TF-IDF model is constructed only once (unless the corpus changes over time) and before any input is given to the system. This phase is also responsible for normalising documents and manipulating the syntagmas according to task-specific heuristics (e.g., filtering out stop words). Specifically, normalisation involves tokenisation, lemmatisation, and stemming[8].

In the second phase, the user input (i.e., the question to answer) is used for answer retrieval. In particular, the answer retriever described in Section 6.1 is modified to consider also the TF-IDF model. Thus, the new answer retriever performs the following steps. First, it converts the input question into an embedding using both the syntagmatic and deep language

[7]Words that are characteristic of the topics discussed by the corpus.

[8]In linguistic morphology, stemming is the process of reducing inflected words to their word stem, base or root form (e.g., the stem of "argue" and "arguing" is "argu").

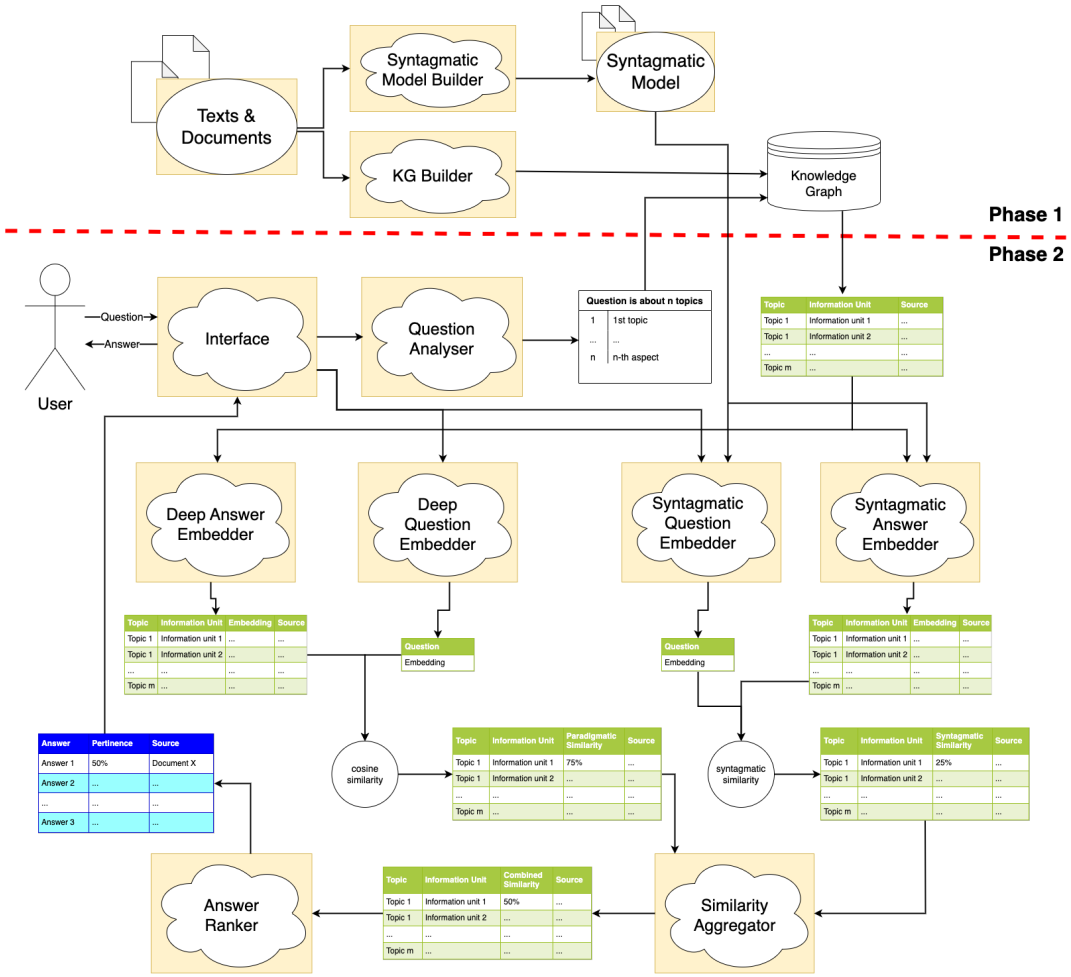


Figure 9.1: ***Flow diagram of SyntagmTuner.*** *Sketch of the pipeline used in our experiments for both text classification and question-answering. Precisely, phase 1 is executed only once when SyntagmTuner is instantiated. This phase mainly consists of creating a syntagmatic (TF-IDF) model. Instead, phase 2 is executed each time the user queries SyntagmTuner. This figure differs from Figure 6.1 in that it involves a syntagmatic model builder, two answer/question embedders and a similarity aggregator.*

models. Then, the question is analysed to identify the topics mentioned by it and extract all the information units about them in the knowledge graph. Next, these information units are embedded with both the syntagmatic and

deep language models. The embeddings are compared to those from the questions to produce the syntagmatic and paradigmatic similarity scores. Similarity scores are then linearly combined to understand which information units have the most similar embedding to the input. Subsequently, the units with the embedding most similar to the input are ranked and returned to the user.

The similarity scores are combined linearly by adding the TF-IDF similarity to the similarity from the deep language model. In particular, both similarities are weighted with task-dependent weights. For example, the query q is embedded by TF-IDF in the vector V_q and by the deep language model in the vector $\bar{V}_q$. Instead, the text fragment p (contained in the corpus) is embedded in the vectors V_p and $\bar{V}_p$. Let s be a similarity function (e.g., the cosine similarity), w_S a default weight for the syntagmatic model and w_P the weight for the deep language model, the similarity $s(V_q, V_p)$ of q with p is S. Instead, the similarity $s(\bar{V}_q, \bar{V}_p)$ is P. Therefore, the final combined similarity between p and q is given by the formula $w_S \cdot S + w_P \cdot P$.

9.3 DiscoLQA: Using Discourse Theory for more Scalable Answer Retrievers

The baseline answer retriever described in Section 6.1 is composed of a pipeline of algorithms for efficient question-answering through the extraction of a knowledge graph from a set of information units. If Hypothesis 5 is correct, it would be possible to specialise such a general-purpose answer retriever to technical languages. This can be done simply by integrating its knowledge graph with external information about the structure of discourse of technical texts without costly training procedures otherwise hampered by the scarcity of data. The overall idea is that using EDUs and AMRs (cf. Section 3.3) as information units for retrieval would help to partly crystallise into the question-answering system the structure of discourse used by technical texts. In other words, it would make the structure of discourse invariant and prevent the answer retriever from using the discourse schemes learned from a common language instead. Therefore, we hereby propose a novel pipeline of algorithms called DiscoLQA, short for Discourse-based Legal Question-Answering, based on Hypothesis 5.

DiscoLQA is composed by the baseline answer retriever described in Section 6.1 extended with a new component responsible for the extraction of special information units representing EDUs and AMRs. In this

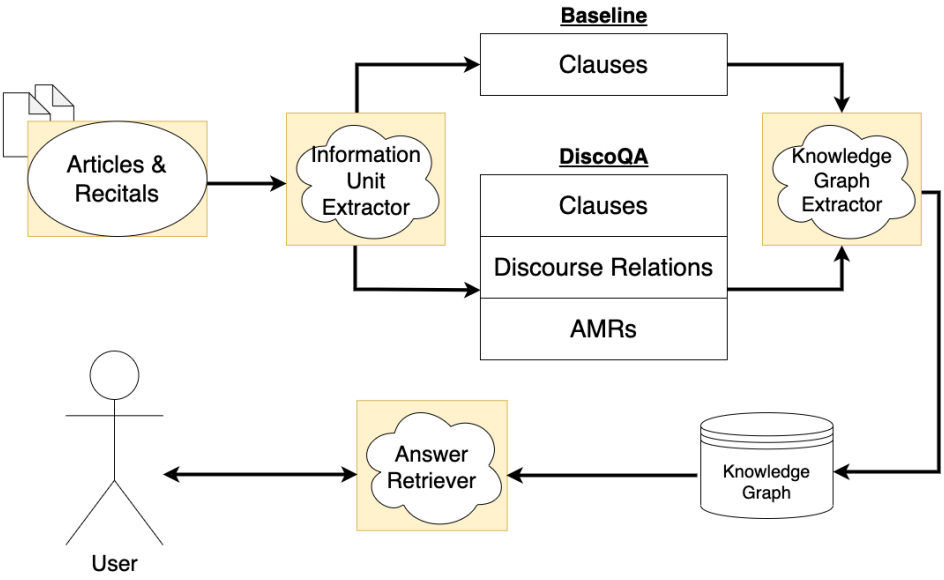


Figure 9.2: ***Sketch of the pipeline used in the baseline and DiscoLQA.*** *The baseline uses only clauses as information units, while DiscoLQA extracts and uses discourse relations and AMRs. Information units are then inputted to the knowledge graph extractor (the template-triplets builder and RDF serializer blocks of Figure 6.2) which then outputs the knowledge base used by the answer retriever.*

sense, the main difference between DiscoLQA and the baseline is (as shown in Figure 9.2) the type of information units considered by the knowledge graph extractor. In particular, the baseline uses as information units all the clauses of the source documents. Instead, DiscoLQA can use as information units not only such clauses but also the AMRs and discourse relations extracted from them.

In other words, DiscoLQA supports more types of information units and allows the retrieval of answers from any combination of clauses, AMRs and discourse relations. Specifically, discourse relations are meant to capture how EDUs are connected, while AMRs are meant to capture the informative components within the EDUs by possibly supporting answering to basic questions such as "`who` did `what` to `whom`, `when` or `where`", as explained in Section 3.3.

Most importantly, by changing the type of information units in DiscoLQA, it is also possible to control the size of the underlying knowledge

graph and the time complexity of the answer retrieval algorithm. Ideally, a smaller knowledge graph with fewer distractors can produce more accurate answers by hiding redundant information units and significantly reduce the time it takes the information retriever to find the correct answers. In other words, DiscoLQA can be used both to increase the accuracy of the question-answering system and to *effectively scale* to larger corpora by reducing the size of the knowledge base used for answering questions.

The AMRs and EDUs used by DiscoLQA are extracted from sentences and paragraphs through a T5-based deep language model[9] pre-trained on a multi-task mixture of unsupervised and supervised tasks. Vanilla T5 is not trained to extract AMRs or EDUs, so we had to fine-tune T5 on some public datasets designed for this task. These datasets are QAMR [135] for the extraction of AMRs and QADiscourse [160] for EDUs and discourse relations. Interestingly, as discussed in Section 3.3, both of these datasets encode AMRs and EDUs as question-answer pairs.

The two considered datasets are tuples of $< s, q, a >$, where s is a source sentence, q is a question (implicitly) expressed in s, and a is an answer expressed in s. So that T5 is fine-tuned to tackle at once the following four tasks per dataset:

1. Extract a given s and q;
2. Extract q given s and a;
3. Extract all the possible q given s;
4. Extract all the possible a given s.

Specifically, we fine-tuned the T5 model on QAMR and QADiscourse for five epochs[10]. The objective of the fine-tuning was to minimise a loss function measuring the difference between the expected output and the output given by T5. A mathematical definition of the loss function is given by Raffel et al. [161].

At the end of the training, the average loss was 0.41, meaning that our fine-tuned T5 model cannot perfectly extract AMRs or EDUs from the text composing the training set. On the one hand, this is a good thing because it is likely that the model did not over-fit on the training set. On the other

[9] T5 is an encoder-decoder model based on the assumption that natural language processing problems can be converted into a text-to-text problem [161].

[10] An epoch is one complete cycle through the entire training dataset

hand, this points to the fact that the AMRs and EDUs extracted by our T5 model can be imperfect, containing errors that could propagate to the answer retrieval system.

9.4 Evaluation of SyntagmTuner and DiscoLQA: a Case Study in European Legislation

As a technical language for the case study and evaluation of SyntagmTuner and DiscoLQA, we chose legal English, also known as legalese.

Legalese is not repetitive. Instead, it adopts vocabulary that is used punctually in particular contexts as if very formal rules govern the sentences they form. It is often canonical and tends to avoid terms with multiple meanings. For example, in legalese, significant fragments tend never to be ambiguous, to have associated definitions, and to make use of combinations of specific nouns and verbs. Applying these formal rules impacts local meanings by constraining the relationships (also known as syntagmatic relations) that words have with others when co-occurring in the writing sequence. Moreover, the classical linguistic structures based on discourse connectives tend to be used differently in law. Legal connectives do not have the same semantic value as everyday discourse. They are operators of deontic rules with multiple meanings (e.g., "*xor*", "*or*", "*and*"). Also, some discourse structures tend not to be used at all because they are not a good practice in legal drafting (e.g., "*but*" and "*for example*").

In other words, the relationship between discourse theory and legalese is complicated and still open to discussion. The application of PDTB to legalese has been explored by some [169, 38], but has yet to have much follow-up. The point is that ordinary discourse theory is better suited to judgments, Hansard reports[11], testimonies and reports of debates. Instead, it seems unsuited to legislative texts and contracts, for which a specific vocabulary (e.g., definitions) or textual structure (e.g., hierarchy) is used to identify meaning through interpretation theory. Indeed, legislative texts have a deeper structure than common sentences. For example, a list has a legal meaning of conditions linked together by specific semantics. Nonetheless, capturing discourse patterns within legal texts can be beneficial for an answer retrieval algorithm such as the one described in Section 6.1.

[11] *Hansard reports* are the transcripts of parliamentary debates in U.K. and many Commonwealth countries.

Considering all these legalese insights in support of Hypotheses 4 and 5, we decided to evaluate both SyntagmTuner and DiscoLQA on Q4EU (cf. Section 9.1). We performed an ablation study to see if they can improve the baseline answer retrieval system described in Section 6.1 without an ad hoc fine-tuning of the deep language models on the downstream task. If those hypotheses are valid, we expect the performance of any generic deep language model to outperform SyntagmTuner and DiscoLQA on technical corpora, such as the European legislative texts of Q4EU.

Notably, the QAMR and QADiscourse datasets used for DiscoLQA are not related to any of the technical domains covered by Q4EU. They do not contain legal documents or text fragments written in legalese. In other words, by fine-tuning T5 on QAMR and QADiscourse, we did not refine T5 on legal texts. Legal fine-tuning would require the costly extraction of a dataset of AMRs and EDUs from legal texts, also considering ad hoc adaptations of discourse theories and abstract meaning representation to legal language.

Specifically, the setup of the experiment is as follows. On the one hand, to test Hypothesis 5, we compare the performance of the baseline answer retriever on the Q4EU dataset with that of DiscoLQA using different combinations of information units. We study the following instances of DiscoLQA:

- **Clause+EDU+AMR**: DiscoLQA which uses clauses, discourse relations and AMRs as information units, all together.
- **Clause+EDU**: DiscoLQA using clauses and discourse relations but not AMRs.
- **Clause+AMR**: DiscoLQA using clauses and AMRs.
- **EDU+AMR**: discourse relations and AMRs.
- **EDU**: discourse relations.
- **AMR**.

Therefore, if one combination of information units performs better than the others, the performance gain can be attributed to the only difference between the instruments: the type of information units adopted. Therefore, if one of the instances of DiscoLQA performs better than the baseline, we would have enough evidence to support Hypothesis 5.

On the other hand, to test Hypothesis 4, we perform the same evaluation on Q4EU, but studying what happens to answer retrievers when they use SyntagmTuner during retrieval. SyntagmTuner and DiscoLQA are compatible because the former is about how the similarity between answers and questions is calculated, and the latter is about the information used to construct the knowledge graph used by the retriever. In particular, for the experiment, we set SyntagmTuner to have $w_S = 0.5$ and $w_P = 0.5$.

The *baseline* is equivalent to SyntagmTuner with $w_S = 0$ and $w_P = 1$, and to DiscoLQA using only clauses as information units. We consider as baseline only the answer retrieval system described in Section 6.1. This is because it is the only system we know of to perform legal question-answering on arbitrary pieces of (English) text without ad hoc fine-tuning or training procedures. We do not have a large enough dataset to train end-to-end question-answering systems on specific European legislation.

To show that the results generalise across different deep language models, we also decided to run the experiment on different state-of-the-art deep neural networks for answer retrieval:

- The Universal Sentence Encoder Q&A model, by TensorFlow [225, Google];
- A variation of MiniLM [219, Microsoft] trained by SBERT [165] on answer retrieval.

We decided to consider only the two models above because they are some of the best available on TensorFlow and SBERT (two state-of-the-art repositories for deep neural networks easily accessible through user-friendly APIs). Unfortunately, we have yet to learn of any deep language model trained specifically on legal answer retrieval. The only exception could be the work by Vold and Conrad [215], though their language model was trained on privacy policies, which are usually written with more plain English than European legislation.

Considering that, with the Q4EU dataset, a single answer is not sufficient[12] to respond to a test query fully, we relied on top-k precision, F1, Normalised Discounted Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR) as evaluation metrics. The top-k precision, or $P@k$, is the fraction of expected answers amongst the top-k retrieved instances. The

[12]DiscoLQA, SyntagmTuner and the baseline have no constraints on the minimum or maximum number of retrievable responses.

top-k F1 score, or $F1@k$, is given by $2\frac{R@k \cdot P@k}{R@k + P@k}$, where the top-k recall, or $R@k$, is measured as the fraction of correct answers retrieved in the top-k instances. In contrast, the top-k NDCG [175] is a measure of ranking quality normalised in $[0, 1]$ that measures the usefulness, or gain, of an answer based on its position in the result list. Instead, the top-k MRR [216] only cares about the single highest-ranked relevant item. It shows what system does the best job at placing a relevant document/passage in to highest rank.

It is important to note that the main difference between precision, F1, MRR and NDCG is that the last two are used to assess the ability of an answer retrieval system to rank correct answers first. Conversely, the other metrics measure how precise and accurate the system is. For these reasons, all selected metrics are considered complementary measurements that may present different lenses into the problem of understanding answer retrieval systems [55].

In Tables 9.6 and 9.7, we show the macro[13] top-k evaluation scores for $k = \{5, 10\}$[14], studying how different types of *information units* and *deep language models* affect answer retrieval with and without SyntagmTuner and DiscoLQA. The evaluation was performed by running the answer retrieval algorithm on all the 6 norms of Q4EU, even though questions in Q4EU usually target only 1 or 2 norms.

These results show that independently of the choice of k, using discourse relations (EDUs) as information units gives the best top-k F1 scores, especially when in combination with clauses and AMRs. DiscoLQA using only discourse relations and AMRs as information units (i.e., *EDU+AMR*) in many cases outperforms the baseline in terms of precision. This happens especially with the Universal Sentence Encoder and with SyntagmTuner, even though the underlying knowledge graph is smaller than the baseline (as shown in Table 9.8). This fact suggests that EDUs and AMRs can retain most of the relevant information of the corpus of technical documents, supporting Hypothesis 5 and helping to create faster and more scalable answer retrievers. Moreover, the fact that the best answer retriever in terms of MRR is overall *Clause+EDU+AMR* further corroborates Hypothesis 5, showing that the considered deep language models tend to be distracted by longer clauses. Indeed, the information within EDUs and AMRs is a sub-

[13] Here, the term "macro" means that precision, F1 and NDCG scores are computed independently for each test query and then averaged, to put an equal weight upon the contribution of each query.

[14] In general, a k greater than or equal to the average number of answers per question (e.g., the score shown in Table 9.1) is recommended.

Table 9.6: ***Q4EU - Scores of Universal Sentence Encoder.*** *This table shows the macro mean (with standard deviation) of the top-k precision (P), F1, NDCG and MRR of each combination of information units, with and without SyntagmTuner. We show the values for $k = \{5, 10\}$. The best column scores are shown in bold, while darker shades of blue indicate higher precision column-wise.*

Universal Sentence Encoder	Top5 Scores		Top10 Scores	
	No TF-IDF (*Baseline*)	SyntagmTuner	No TF-IDF (*Baseline*)	SyntagmTuner
Clause (*Baseline*)	P: 0.385 ± 0.325 F1: 0.28 ± 0.212 NDCG: 0.301 ± 0.234 MRR: 0.529 ± 0.42	P: 0.516 ± 0.323 F1: 0.383 ± 0.202 NDCG: 0.425 ± 0.25 MRR: 0.675 ± 0.373	P: 0.504 ± 0.339 F1: 0.252 ± 0.176 NDCG: 0.298 ± 0.219 MRR: 0.546 ± 0.401	P: 0.614 ± 0.31 F1: 0.331 ± 0.166 NDCG: 0.412 ± 0.226 MRR: 0.686 ± 0.355
AMR	P: 0.318 ± 0.321 F1: 0.247 ± 0.239 NDCG: 0.301 ± 0.305 MRR: 0.478 ± 0.446	P: 0.434 ± 0.309 F1: 0.322 ± 0.209 NDCG: 0.397 ± 0.314 MRR: 0.582 ± 0.417	P: 0.401 ± 0.34 F1: 0.218 ± 0.196 NDCG: 0.304 ± 0.292 MRR: 0.489 ± 0.436	P: 0.537 ± 0.339 F1: 0.29 ± 0.186 NDCG: 0.417 ± 0.314 MRR: 0.587 ± 0.411
EDU	P: 0.454 ± 0.345 F1: 0.326 ± 0.241 NDCG: 0.361 ± 0.288 MRR: 0.59 ± 0.419	P: 0.518 ± 0.325 F1: 0.364 ± 0.205 NDCG: 0.412 ± 0.268 MRR: 0.677 ± 0.366	**P: 0.596 ± 0.342** **F1: 0.304 ± 0.192** **NDCG: 0.366 ± 0.253** MRR: 0.607 ± 0.397	P: 0.635 ± 0.296 F1: 0.353 ± 0.191 NDCG: 0.434 ± 0.245 MRR: 0.686 ± 0.349
EDU+AMR	P: 0.459 ± 0.337 F1: 0.332 ± 0.226 NDCG: 0.379 ± 0.301 MRR: 0.588 ± 0.424	P: 0.528 ± 0.32 F1: 0.383 ± 0.204 **NDCG: 0.442 ± 0.279** MRR: 0.694 ± 0.371	P: 0.569 ± 0.339 F1: 0.299 ± 0.201 NDCG: 0.364 ± 0.279 MRR: 0.603 ± 0.405	P: 0.649 ± 0.299 F1: 0.359 ± 0.186 **NDCG: 0.439 ± 0.244** MRR: 0.703 ± 0.356
Clause+AMR	P: 0.422 ± 0.313 F1: 0.318 ± 0.224 NDCG: 0.359 ± 0.269 MRR: 0.603 ± 0.423	**P: 0.547 ± 0.324** **F1: 0.396 ± 0.203** NDCG: 0.423 ± 0.25 MRR: 0.682 ± 0.358	P: 0.555 ± 0.331 F1: 0.285 ± 0.194 NDCG: 0.345 ± 0.244 **MRR: 0.618 ± 0.404**	P: 0.63 ± 0.312 F1: 0.352 ± 0.19 NDCG: 0.421 ± 0.234 MRR: 0.685 ± 0.353
Clause+EDU	**P: 0.467 ± 0.353** F1: 0.34 ± 0.243 NDCG: 0.356 ± 0.262 MRR: 0.592 ± 0.415	P: 0.535 ± 0.339 **F1: 0.396 ± 0.226** NDCG: 0.426 ± 0.26 MRR: 0.688 ± 0.38	P: 0.572 ± 0.348 F1: 0.291 ± 0.197 NDCG: 0.334 ± 0.239 MRR: 0.605 ± 0.399	**P: 0.666 ± 0.301** **F1: 0.368 ± 0.187** NDCG: 0.417 ± 0.222 MRR: 0.699 ± 0.361
Clause+EDU+AMR	P: 0.457 ± 0.328 **F1: 0.348 ± 0.234** **NDCG: 0.381 ± 0.288** **MRR: 0.604 ± 0.413**	P: 0.526 ± 0.323 F1: 0.39 ± 0.215 NDCG: 0.425 ± 0.258 **MRR: 0.697 ± 0.37**	P: 0.586 ± 0.332 F1: 0.303 ± 0.201 NDCG: 0.358 ± 0.264 **MRR: 0.618 ± 0.394**	P: 0.665 ± 0.293 **F1: 0.368 ± 0.185** NDCG: 0.419 ± 0.221 **MRR: 0.705 ± 0.357**

set of the information within the set of clauses composing the knowledge graph of the baseline.

Furthermore, as expected, we can see that the precision of SyntagmTuner exceeds the baseline in all cases (especially with the Universal Sentence Encoder). This supports Hypothesis 4, suggesting that general-purpose deep language models may have difficulty capturing syntagmatic relations within technical texts such as the European legislative texts. Furthermore, evidence gathered in [200, 193] indicates that Hypothesis 4 also applies to other technical texts, i.e., UN General Assembly resolutions.

Table 9.7: ***Q4EU - Scores of MiniLM.*** *For further details on interpreting this table, read the caption of Table 9.6.*

MiniLM	Top5 Scores		Top10 Scores	
	No TF-IDF (*Baseline*)	SyntagmTuner	No TF-IDF (*Baseline*)	SyntagmTuner
Clause (*Baseline*)	P: 0.524 ± 0.343 F1: 0.4 ± 0.239 NDCG: 0.457 ± 0.29 MRR: 0.697 ± 0.391	P: 0.539 ± 0.318 F1: 0.415 ± 0.207 **NDCG: 0.478 ± 0.257** MRR: 0.741 ± 0.344	P: 0.627 ± 0.296 F1: 0.347 ± 0.18 NDCG: 0.427 ± 0.26 MRR: 0.707 ± 0.374	P: 0.643 ± 0.3 F1: 0.372 ± 0.187 NDCG: 0.453 ± 0.225 MRR: 0.746 ± 0.335
AMR	P: 0.432 ± 0.325 F1: 0.332 ± 0.236 NDCG: 0.412 ± 0.32 MRR: 0.605 ± 0.43	P: 0.456 ± 0.324 F1: 0.34 ± 0.222 NDCG: 0.423 ± 0.318 MRR: 0.625 ± 0.418	P: 0.511 ± 0.345 F1: 0.29 ± 0.207 NDCG: 0.424 ± 0.31 MRR: 0.613 ± 0.42	P: 0.56 ± 0.333 F1: 0.308 ± 0.192 NDCG: 0.447 ± 0.305 MRR: 0.631 ± 0.409
EDU	P: 0.454 ± 0.35 F1: 0.328 ± 0.233 NDCG: 0.379 ± 0.295 MRR: 0.596 ± 0.415	P: 0.492 ± 0.321 F1: 0.365 ± 0.214 NDCG: 0.423 ± 0.281 MRR: 0.667 ± 0.385	P: 0.531 ± 0.365 F1: 0.286 ± 0.198 NDCG: 0.366 ± 0.272 MRR: 0.602 ± 0.407	P: 0.628 ± 0.308 F1: 0.348 ± 0.19 NDCG: 0.439 ± 0.256 MRR: 0.677 ± 0.369
EDU+AMR	P: 0.478 ± 0.342 F1: 0.371 ± 0.243 NDCG: 0.417 ± 0.295 MRR: 0.634 ± 0.415	P: 0.527 ± 0.319 F1: 0.394 ± 0.212 NDCG: 0.454 ± 0.277 MRR: 0.705 ± 0.37	P: 0.587 ± 0.346 F1: 0.323 ± 0.201 NDCG: 0.395 ± 0.26 MRR: 0.645 ± 0.399	**P: 0.688 ± 0.292** F1: 0.38 ± 0.185 **NDCG: 0.459 ± 0.243** MRR: 0.713 ± 0.355
Clause+AMR	**P: 0.529 ± 0.337** F1: 0.407 ± 0.235 **NDCG: 0.465 ± 0.287** MRR: 0.709 ± 0.386	P: 0.54 ± 0.317 F1: 0.412 ± 0.204 NDCG: 0.471 ± 0.256 MRR: 0.741 ± 0.334	P: 0.613 ± 0.318 F1: 0.341 ± 0.192 NDCG: 0.426 ± 0.266 MRR: 0.715 ± 0.376	P: 0.652 ± 0.3 F1: 0.375 ± 0.189 NDCG: 0.451 ± 0.224 MRR: 0.742 ± 0.33
Clause+EDU	P: 0.512 ± 0.361 F1: 0.401 ± 0.26 NDCG: 0.46 ± 0.304 MRR: 0.708 ± 0.408	**P: 0.562 ± 0.314** **F1: 0.434 ± 0.197** NDCG: 0.471 ± 0.243 MRR: 0.728 ± 0.341	**P: 0.638 ± 0.312** F1: 0.349 ± 0.181 NDCG: 0.425 ± 0.256 MRR: 0.726 ± 0.379	P: 0.679 ± 0.287 F1: 0.386 ± 0.182 NDCG: 0.449 ± 0.218 MRR: 0.731 ± 0.334
Clause+EDU+AMR	P: 0.522 ± 0.355 **F1: 0.408 ± 0.252** NDCG: 0.463 ± 0.295 **MRR: 0.715 ± 0.4**	P: 0.549 ± 0.316 F1: 0.421 ± 0.2 NDCG: 0.472 ± 0.244 **MRR: 0.752 ± 0.33**	P: 0.637 ± 0.319 **F1: 0.356 ± 0.188** **NDCG: 0.432 ± 0.256** **MRR: 0.729 ± 0.377**	P: 0.683 ± 0.289 **F1: 0.39 ± 0.181** NDCG: 0.457 ± 0.214 **MRR: 0.756 ± 0.322**

Finally, despite their differences, both MiniLM (the best) and the Universal Sentence Encoder behave similarly, suggesting that the information units we considered and SyntagmTuner play a role that is independent of the underlying general-purpose language model used for retrieval. Overall, these findings support our hypotheses. They show that it is possible to significantly improve a general-purpose language model, making it perform better with legal texts. This is possible by better capturing syntagmatic relationships and using noiseless information units, i.e., decomposing a generic clause into one or more discourse relations or AMRs.

In other words, as expected, the information units representing the (generic) clauses carry enough noise to distract the answer retriever. By breaking the sentences into EDUs and explicitly keeping their relations, we can crystallise the discourse structure into the knowledge graph, making it invari-

Table 9.8: ***Q4EU - Knowledge graphs size.*** *This table shows how the number of unique retrievable triplets in the knowledge graphs used for question-answering change when changing information units.*

	Clause (*Baseline*)	AMR	EDU	EDU+AMR	Clause+AMR	Clause+EDU	Clause+EDU+AMR
Unique Retrievable Triplets	28,718	5,831	22,393	26,139	32,642	42,250	45,775

ant. Therefore the answer retriever is forced to "reason" over the discourse patterns, minimising the chances of relying on common-sense discourse schemes instead.

Examples of how EDUs and AMRs are important for some questions of the Q4EU dataset are shown in Table 9.9. A qualitative analysis of the responses produced by the algorithm shows that it can identify useful normative references to ensure the completeness of the answer and develop an overview. For instance, among the answers to the question "Who decides precedence in the event of a conflict between a European arrest warrant and a request for extradition from a third country?" the algorithm identifies Article 16.3 (the most relevant answer) and suggests Recital 8, which helps interpret Article 16.3. Furthermore, for the same question, the algorithm also suggests Article 10.6, which, while not suitable for answering the question, leads the jurist to complementary points of reference for more holistic reasoning and interpretation.

Both Tables 9.9 and 9.10 show errors committed by the answer retrievers and the extractor of information units. These examples indicate at least two different types of errors. The first type occurs when an information unit is not correctly extracted so that it is semantically or grammatically incorrect, such as in the first and fourth rows of Table 9.9. This type of error is not particularly serious since, in some cases, the underlying language model is resistant to inaccuracies[15], still allowing a correct answer to be retrieved, as shown in Table 9.9.

This first type of error is usually caused by the automatic extraction of AMRs and EDUs by a neural network, as described in Section 9.3. For this reason, in both Tables 9.9 and 9.10, there are examples of information units that do not perfectly overlap with the "answer text". On the contrary, the second type of error is due to mistakes in the deep language model for

[15]This mainly happens because the answer retriever concatenates the information units with their context, as explained at the end of Section 6.1.

Table 9.9: ***Q4EU - Examples of correct answers.*** *This table shows a few examples of answers correctly given by the baseline and DiscoLQA (EDU+AMR). For each answer, we provide the key information unit (IU for short) used by the retriever to associate the correct answer to the question. Nonetheless, even if the answer is selected correctly, the reason for selecting it might be wrong. Thus, errors in the information units are shown with a strike-through.*

Question	Algorithm	Answer	Answer Text	Answer Pertinence	Key IU	Type of IU
What court has jurisdiction in case of a counter-claim?	EDU+ AMR	B Art. 8.3	A person domiciled in a Member State may also be sued: (3) on a counter-claim arising from the same contract or facts on which the original claim was based, in the court in which the original claim is pending;	49.35%	~~In what manner~~ may a person domiciled in a Member State be sued? in the court in which the original claim is pending	Discourse Relation
Where can an employee sue their employer?	EDU+ AMR	B Art. 23.2	which allows the employee to bring proceedings in courts other than those indicated in this Section.	25.81%	What allows the employee to bring proceedings? courts	AMR
Who determines disputes under a contract?	EDU+ AMR	B Art. 17.2	Where a consumer enters into a contract with a party who is not domiciled in a Member State but has a branch, agency or other establishment in one of the Member States, that party shall, in disputes arising out of the operations of the branch, agency or establishment, be deemed to be domiciled in that Member State.	35.26%	Who enters into a contract with a party? a consumer	AMR
What court has jurisdiction in case of a counter-claim?	EDU+ AMR	B Art. 14.2	The provisions of this Section shall not affect the right to bring a counter-claim in the court in which, in accordance with this Section, the original claim is pending.	59.30%	~~In what case~~ is the right to bring a counter-claim? in the court	Discourse Relation
What factors should be taken into account for conferring the jurisdiction to determine disputes under a contract?	Baseline	B Art. 25.5	An agreement conferring jurisdiction which forms part of a contract shall be treated as an agreement independent of the other terms of the contract.	34.41%	the terms of the contract	Clause
What kind of agreement between parties are regulated by these Regulations?	Baseline	RI Art. 1.2.e	The following shall be excluded from the scope of this Regulation: (e) arbitration agreements and agreements on the choice of court;	37.18%	~~the scope of this Regulation~~	Clause

Table 9.10: ***Q4EU - Examples of wrong answers.*** *This table shows a few examples of answers wrongly given by the baseline and DiscoLQA (EDU+AMR). For more details on how to read this table, read the caption of Table 9.9.*

Question	Algorithm	Answer	Answer Text	Answer Pertinence	Key IU	Type of IU
What kind of agreement between parties is regulated by these Regulations?	Baseline	B Art. 73.3	This Regulation shall not affect the application of bilateral conventions and agreements between a third State and a Member State concluded before the date of entry into force of Regulation (EC) No 44/2001 which concern matters governed by this Regulation.	45.16%	of conventions and agreements	Clause
When are two actions to be considered related according to Regulation Brussels I Bis?	EDU+ AMR	B Art. 71.2.a	The court hearing the action shall, in any event, apply Article 28 of this Regulation;	27.53%	In what case shall the court hearing the action apply Article 28 of the Regulation? In any event	Discourse Relation
Can the parties choose a different applicable law for different parts of the contract?	EDU+ AMR	R1 Rec. 14	Should the Community adopt, in an appropriate legal instrument, rules of substantive contract law, including standard terms and conditions, such instrument may provide that the parties may choose to apply those rules.	41.28%	What may the parties choose to apply? substantive contract law	AMR

answer retrieval. As shown in Table 9.10, this type of error can be rather severe, causing wrong answers to be selected by the retriever.

We also studied how (top10) precision scores vary when the *context specificity* changes. The results partly confirmed our expectations. We can see a trend where average top10 precision increases proportionally to the context specificity. This is clear in almost all instances of DiscoLQA (with SyntagmTuner) and shown in Table 9.11.

Our expectations are based on the fact that:

- The specificity of a question is low when it asks something that cannot be explicitly found in the Regulations but requires a holistic analysis of principles, competence rules, and so forth;
- Questions with low specificity usually tend to have more expected answers, and it may be harder to find all of them;

Table 9.11: ***Q4EU - P@10 by context specificity.*** *Mean top10 precision scores (with standard deviation) grouped by context specificity, for MiniLM with SyntagmTuner. The best column results are in bold, while darker shades of blue indicate higher precision column-wise.*

P@10 on MiniLM with SyntagmTuner	Specificity		
	High	Normal	Low
Clause (*Baseline*)	0.772 ± 0.304	0.62 ± 0.301	0.542 ± 0.244
AMR	0.52 ± 0.387	0.612 ± 0.327	0.535 ± 0.267
EDU	0.764 ± 0.291	0.599 ± 0.306	0.527 ± 0.277
EDU+AMR	0.798 ± 0.251	**0.677 ± 0.321**	**0.593 ± 0.254**
Clause+AMR	0.749 ± 0.304	0.661 ± 0.307	0.542 ± 0.244
Clause+EDU	**0.821 ± 0.245**	0.655 ± 0.301	0.567 ± 0.248
Clause+EDU+AMR	**0.821 ± 0.245**	0.667 ± 0.306	0.567 ± 0.248

Table 9.12: ***Q4EU - Percentage of answers more/less precise than the baseline.*** *Percentage of queries for which DiscoLQA (with SyntagmTuner and MiniLM) made a positive/negative difference from the baseline in terms of top10 precision. Percentages are grouped by context specificity. The best column deltas are in bold, while darker shades of blue indicate higher positive deltas (the difference between "more" and "less") column-wise.*

P@10 on MiniLM with SyntagmTuner	Specificity		
	High	Normal	Low
AMR	More: 13.64% Less: 45.45%	More: 21.43% Less: 17.86%	More: 13.64% Less: 18.18%
EDU	More: 13.64% Less: 13.64%	More: 14.29% Less: 21.43%	More: 22.73% Less: 22.73%
EDU+AMR	More: 9.09% Less: 9.09%	**More: 25.0%** **Less: 7.14%**	**More: 22.73%** **Less: 9.09%**
Clause+AMR	More: 0.0% Less: 4.55%	**More: 17.86%** **Less: 0.0%**	More: 0.0% Less: 0.0%
Clause+EDU	**More: 9.09%** **Less: 4.55%**	More: 14.29% Less: 3.57%	More: 9.09% Less: 4.55%
Clause+EDU+AMR	**More: 9.09%** **Less: 4.55%**	**More: 21.43%** **Less: 3.57%**	More: 9.09% Less: 4.55%

- Multi-hop reasoning is usually required to answer questions with low specificity, but the considered answer retrievers are not equipped for that kind of reasoning (yet).

For example, the question "How should a contract be interpreted according to Regulation Rome I?" has very low specificity. It requires pinpointing both recitals and articles for a proper answer, therefore, more distinct and distant paragraphs. Most of the questions regarding hermeneutics would probably require a broader view of the subject, having a low specificity to the Regulation, therefore requiring multi-hop reasoning.

Furthermore, we also show in Table 9.12 the percentage of queries for which the best instance of DiscoLQA made a positive/negative difference from the baseline in terms of top10 precision grouped by specificity. In particular, EDU+AMR (the version of DiscoLQA using only AMRs and EDUs) was able to produce 19.4% more precise top10 answers than the baseline, using MiniLM and SyntagmTuner. This percentage rises to 22.7% and 25% when considering only questions with low and normal specificity, respectively.

We tested and evaluated DiscoLQA on specific European legislative texts and a relatively small dataset without comparing our results with deep language models pre-trained on legal corpora. Nonetheless, even though Q4EU is about different legal sub-domains (respectively: Private International Law, the European arrest warrant, data protection and electronic signatures), our instances of DiscoLQA and SyntagmTuner were able to generalise well across them, exceeding the baselines in all cases.

Notably, that happened even though we built DiscoLQA and SyntagmTuner to perform zero-shot question-answering without any training procedure involving European legislation or (more generally) legal documents. Therefore, in practice, DiscoLQA and SyntagmTuner can potentially be deployed on a wide variety of domains for which data scarcity is unavoidable (e.g., the use cases considered for showcasing our YAI). In particular, for deploying DiscoLQA and SyntagmTuner, one would not need to manually create any new time-consuming dataset like Q4EU.

As future work, we point to the possibility of specialising the algorithm for the extraction of EDUs and AMRs to legislative texts or other technical texts, taking into account what we already know about (legal) connectors and discourses.

CHAPTER 10

How to Improve the Explanatory Power of an Intelligent Textbook: a Case Study in Legal Writing

In the previous chapters, we discussed how YAI could help produce more usable explanations of the software documentation of AI systems. However, if the theory expounded in Part I is correct, YAI should also be usable for education. As pointed out by UNESCO, the United Nations specialised agency for education, in one of its recent publications [134], the opportu-

The work presented in Chapter 10 was partially supported by the European Union's Horizon 2020 research and innovation programme under the MSCA grant agreement No 777822 "GHAIA: Geometric and Harmonic Analysis with Interdisciplinary Applications". It was developed in collaboration with prof. Kevin Ashley and prof. Peter L. Brusilovsky, from the University of Pittsburgh. *F. Sovrano*: conceptualization, methodology, software, data curation, original draft preparation, visualization, investigation, validation, formal analysis. *K. Ashley*: conceptualization, review, editing and supervision. *P. L. Brusilovsky*: supervision. We thank the copyright holders of [35] for allowing us to use (parts of) the book to conduct the experiments, carry out the case study and present this research work.

nities and challenges that AI offers for education in the AI era are yet to be fully understood. For this reason, in this chapter, we examine applications of YAI to increase the *explanatory power* of educational textbooks. In particular, we consider a case study in the legal domain related to the United States' legal system. Therefore, we extend the work done for YAI4Hu (cf. Chapter 6), employing the question-answering strategies explained in Chapter 9 and proposing new heuristics to improve the *illocutionary force* of the YAI system.

Empirical evidence gathered in Chapter 7 suggests that explanatory illocution consists of answering implicit (archetypal) questions. So, if Hypothesis 1 (cf. Section 3.2) is true (as it seems), it follows that the explanatory illocution of a YAI system can be improved by adequately identifying those implicit questions that are the most interesting for the explainee. Though, anticipating these questions is not a trivial task, not even for the explainee.

Nonetheless, assuming that the content of a textbook (or any other collection of texts) properly explains a given explanandum, then:

- Any question falling outside the scope of the collection of documents could not be answered, thus not being useful for the explainee;
- Whoever wrote the textbook tried to explain as well as possible (for her/his narrative purposes) the most important topics at hand, thus, according to the adopted definition of explanation (cf. Chapter 3), implicitly identifying the most important questions whose answers provide a good overview of the topics.

Therefore, we make the following hypothesis:

Hypothesis 6. *The most useful implicit questions a user may have about a collection of texts are those best answered by the whole collection. These questions are neither too detailed (because they would otherwise only be answered in a minor part of the collection) nor too general (because they would be answered inaccurately in the more detailed textual content).*

Throughout this chapter, we will discuss how to leverage Hypothesis 6 to identify the questions best answered by a collection of texts, thus algorithmically organising educational explanations accordingly as intelligent overviews. As a *case study* for our proposed YAI for education (YAI4Edu, for short), we considered a teaching scenario where the excerpts of a textbook, "*United States Legal Language and Culture: An Introduction to the*

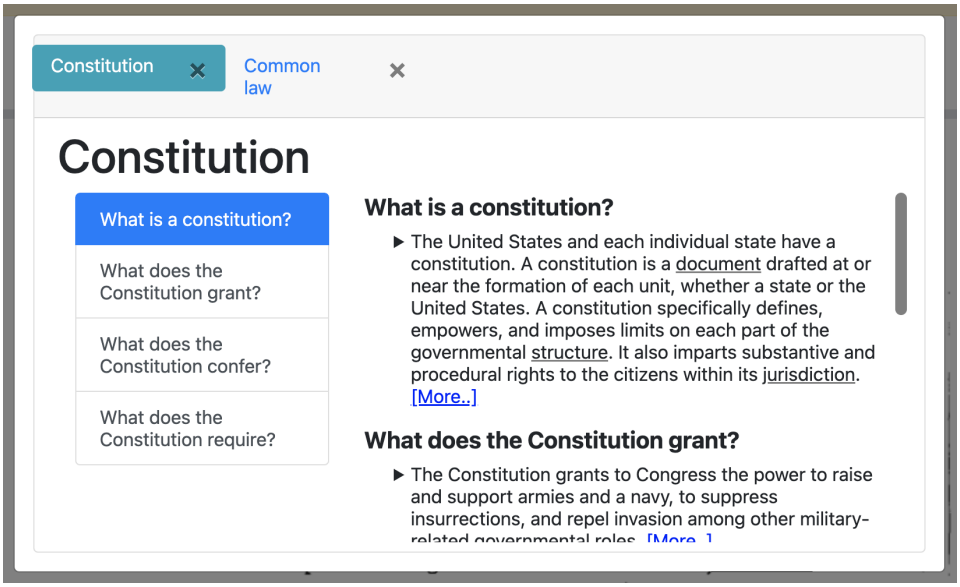


Figure 10.1: ***Example of intelligent explanatory overview generated by YAI4Edu.*** *This figure contains an example of interactive overview in the form of a scrollspy showing relevant questions and answers as explanations. The reader can select a new topic to overview by clicking on any underlined word.*

U.S. Common Law System" [35], together with the encyclopaedia of the *Legal Information Institute of the Cornell Law School* and thousands of cases of the Board of Veterans' Appeals (BVA)[1] are used for teaching how to write a legal memorandum[2] in support of a disability claim for Post-Traumatic Stress Disorder (PTSD) according to the U.S. legal system. For an example of an intelligent explanation generated by YAI4Edu in this scenario, see Figure 10.1.

As an *example* to clarify what Hypothesis 6 means in this case, let us suppose we want to explain what a legal memorandum is. The selected textbook [35] does it by describing what a memo is in a legal sense, what it is for, what the proper form of a legal memorandum is and what sections it should include. The textbook also provides *secondary details*, explain-

[1]The BVA is an administrative tribunal within the United States Department of Veterans Affairs that determines whether U.S. military veterans are entitled to claimed veterans' benefits and services.

[2]A legal memorandum is an organised document that summarises relevant laws to support a conclusion on a particular legal issue.

ing each step of drafting a memorandum, why writing a memo is difficult, what the heading of a memorandum contains, and so on. Hence, in this case, Hypothesis 6 implies that the most useful implicit questions to ask about a legal memorandum for the textbook are not those whose answers are only secondary details. This is because they are too specific to represent the textbook's explanatory content adequately. Instead, the best choices are the primary questions such as "what is the proper form of a legal memorandum", "what is a memo in a legal sense", because they best represent the content of the textbook.

To evaluate YAI4Edu and verify Hypothesis 6, we conducted a within-subjects *user study*, comprising more than 100 students. This was done to study how different strategies to identify helpful implicit questions impact the quality of the resulting explanations. In particular, during the study, different explanations were given to English-speaking students about the task of *writing a legal memorandum*.

We compared the explanations generated by the overview generator of YAI4Edu (called *Intelligent Explanation Generator*; relying on Hypothesis 6) with those of the following two baseline algorithms:

- A *random explanation generator*: an algorithm that organises explanations by *randomly* selecting implicit questions from those answered by the corpus of considered texts;
- A *generic explanation generator*: the explainer adopted by YAI4Hu to generate overviews (see Section 6.3), which uses very *generic questions* (e.g., `why`, `how`, `what`) instead of questions extracted from the textbook, under the assumption that all possible (implicit) questions are instances of such generic questions.

The following sections will describe the differences between YAI4Edu and YAI4Hu, how the Intelligent Explanation Generator works, the case study and experiment considered, and discuss the empirical results and limitations. We also release the source code of YAI4Edu[3] and the anonymised data collected to evaluate it under MIT license at `https://github.com/Francesco-Sovrano/YAI4Edu`.

[3]We cannot release the textbook excerpts [35] because they are copyrighted.

10.1 YAI4Edu: a YAI for Improving the Explanatory Power of an Intelligent Textbook

Some studies suggest that using intelligent textbooks[4] and interactive e-books leads to an increase in use, motivation, and learning gains versus static e-books [70]. Of the several streams of work on the topic of interactive e-books and intelligent textbooks, most focus on the cognitive process of the reader, studying how to enhance the pedagogical productivity of textbooks through expert systems or sophisticated interfaces. They usually accomplish this by:

- Showing personal progress through open learner models [107, 108];
- Specialising on ad hoc tasks through some domain modelling [18, 58, 45, 130];
- Modelling a student through questions, in order to identify and suggest personalised contents [209, 141, 130];
- Associating pedagogically valuable quizzes and exercises to portions of the e-book [222, 184, 40, 41];
- Providing tools for manually creating new interactive e-books [218, 159, 111].

The use of AI for the automatic generation of interactive e-books seems to be under-explored.

The benefit of answering questions for learning has been shown in many studies [168, 153], further supporting the assertion that explaining is akin to question-answering and that organising contents on a question-answer base might be beneficial for the explainee. However, creating questions with a proper level of detail that effectively helps students' learning usually requires experience and extensive efforts [184]. Hence, we hereby propose YAI4Edu, an extension of YAI4Hu (cf. Chapter 10) to automatically transform static educative e-books (in PDF, XML or HTML format) into interactive intelligent textbooks by increasing their explanatory power.

In contrast to all the previously mentioned literature examples, we investigate how to use YAI to fully automatically enhance (static) educational

[4]Intelligent textbooks extend regular textbooks by integrating machine-manipulable knowledge [218].

books by making them interactive, thus reducing the sparsity of relevant information and increasing the *explanatory power* of the medium. More specifically, we are not interested in the task of generating verbatim[5] questions for quizzes or exercises as [222, 184, 40, 41], but instead, we pursue the different idea that questions (even non-verbatim ones) can be an effective criterium to organise and categorise the content of explanations. Moreover, instead of considering any question as a suitable candidate for this task, we empirically show that some questions are more helpful than others and that the best questions for explanatory overviews are neither too generic nor too specific.

The differences between YAI4Edu and YAI4Hu are mainly three. First, for answer retrieval, YAI4Edu relies on SyntagmTuner (see Section 9.2) and DiscoLQA (Clause+EDU+AMR; see Section 9.3), using MiniLM as pre-trained deep language model (the best, according to the empirical results discussed in Section 9.4). SyntagmTuner and DiscoLQA are likely to produce better explanations considering that, as previously anticipated and described in the following subsections, the chosen case study is about legal writing and legal English. Secondly, YAI4Edu relies on a more advanced mechanism for producing overviews, called *intelligent overviewing* (an example of intelligent overview is shown in Figure 10.1). Intelligent overviewing is designed to extract the most helpful questions a textbook (or supplementary text) is answering by exploiting parts of the DiscoLQA pipeline. Third, YAI4Edu uses a more advanced algorithm for identifying which words are to be overviewed, called *smart annotation generator*, automatically identifying a glossary of words representing the most explained textbook contents.

Assuming that the goal of a textbook is to explain something to the reader, and based on the theoretical understandings expressed in Chapter 3, our YAI4Edu is designed around the idea that organising the explanatory space (i.e., the space of all possible bits of explanation) as clusters of archetypal questions and answers is beneficial for an explainee. In particular, as shown in the flow chart of Figure 10.2, YAI4Edu uses the following **predefined interactions inherited from YAI4Hu** to allow the user to explore this explanatory space:

- **Open-ended question-answering**: the user writes a question, and then it gets one or more relevant punctual answers;

[5]The verbatim question is a question for which an answer can be literally identified in a related instructional text (i.e., source text) [184].

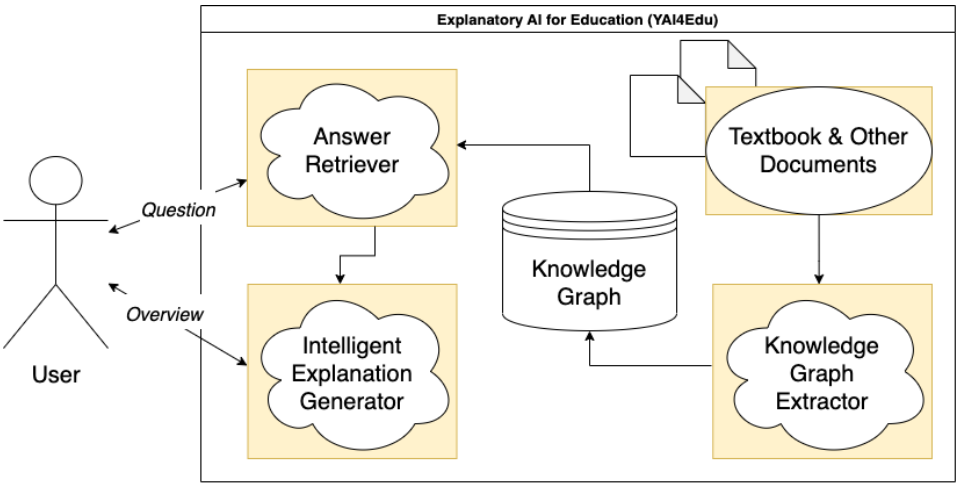


Figure 10.2: ***Simplified flow diagram of YAI4Edu.*** *This diagram shows the main components of YAI4Edu. As in YAI4Hu, the user can ask questions and get overviews. However, differently, YAI4Edu uses a new component for generating overviews, as described in Section 10.1.*

- **Aspect overviewing**: the user selects an aspect of the explanandum (i.e., contained in an answer), receiving as explanation a set of *relevant* answers to archetypal questions involving other different aspects that can be explored as well. Answers can also be expanded, increasing the level of detail.

In other words, interaction is given by: *i) word glosses* that can be clicked to open an overview, *ii)* a special kind of *search box* that allows the reader to get answers about any open-ended English question.

As hypothesised (cf. Hypothesis 6), archetypal questions that are too generic are unlikely to represent the explanatory goals of a sufficiently complex and elaborated collection of texts. The archetypal questions originally used by YAI4Hu for *overviewing* are too generic and predefined, frequently not adhering to the explanatory requirements of the overview. Therefore, considering the need for YAI4Edu to be pedagogically helpful, we designed a novel theoretically grounded AI algorithm able to quantify how much an archetypal question is likely to be representative of the explanatory goals of a collection of texts. We called this algorithm the *Intelligent Explanation Generator*.

Instead of considering only predefined generic archetypal questions,

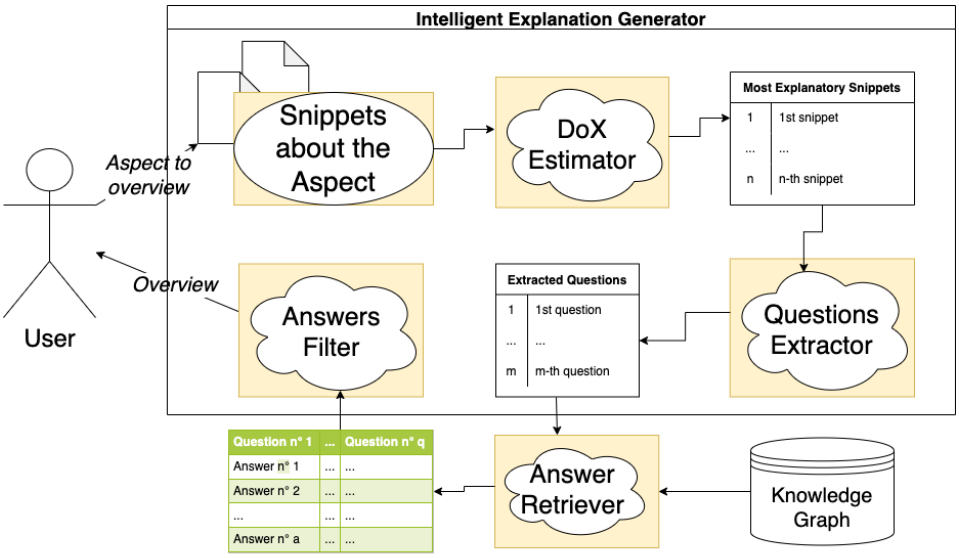


Figure 10.3: ***Flow diagram of the Intelligent Explanation Generator.*** *This diagram shows how an explainee can obtain an intelligent overview. First, the user decides which aspect of the explanandum to overview (i.e., by clicking on an annotated word available on the screen). Then the system extracts an explanation from the textbook or any other collection of texts (e.g., other textbooks, an encyclopedia) by using an AI algorithm for question extraction and an AI algorithm for answer retrieval as described in Section 6.1.*

the Intelligent Explanation Generator also considers more domain-specific (archetypal) questions automatically identified by the AI for question-answer extraction used by DiscoLQA and described in Section 9.3. More specifically, the **workflow of the Intelligent Explanation Generator** consists of the following three steps (summarised in Figure 10.3).

Step 1. The algorithm computes the DoX of all snippets of text about a given explanandum (i.e., an aspect to overview), finding the top k snippets with the highest DoX and finding also the archetypal questions extracted from them by the algorithm described in Section 9.3;

Step 2. The algorithm identifies a set of pertinent answers within the text snippets for each question selected in the previous step. An answer is

said to be pertinent to a question when its pertinence score[6] is greater than a given pertinence threshold p;

Step 3. The algorithm filters the pertinent answers, keeping the best q questions and a answers by executing the following sub-steps: *i)* questions that are too long are removed, i.e., questions whose length (without considering the length of the explanandum label) exceeds a threshold L; *ii)* if a question has some grammatical error, it is automatically corrected via Gramformer[7], a deep neural network; *iii)* questions that are too similar are removed,[8] prioritising the questions extracted from the most explanatory snippets (i.e., those with the highest DoX) and the shortest questions; *iv)* answers that are too short or too long are removed; *v)* the questions with no valid answers are removed; *vi)* the answers that could be assigned to several questions are given to the question with the highest estimated pertinence; *vii)* for each question, only the a answers with the highest pertinence score are kept; *viii)* the questions are sorted by the decreasing pertinence of the first answer, and only the top q questions are kept.

Importantly, step 1 is performed before step 2 to reduce the asymptotic time complexity of step 2. Selecting the questions best answered by the corpus (step 2) has an asymptotic complexity $O(|Q_c| \cdot |S|)$ that grows with the number of questions extracted from the snippets of text, where Q is the set of questions about an aspect c to be explained and S is the set of snippets of text. Therefore, this complexity in the worst-case scenario (without step 1) can be quadratic in the size of the textbook or collection of texts, i.e., $O(|S|^2)$.

Rather than having a quadratic complexity, a computationally simpler approach can perform an initial filtering procedure to consider only those questions coming from the paragraphs with the highest DoX (as step 1 does), thus converting $|Q_c|$ into a constant number independent from $|S|$. Hence, considering that computing the DoX of $|S|$ snippets of text has an asymptotic time complexity equal to $O(|S|)$, it follows that step 1 reduces the complexity of the Intelligent Explanation Generator to $O(|S|)$.

[6]Pertinence scores are numbers in $[0, 1]$ computed by measuring the cosine similarity between vectorial representations of question and answer obtained through deep neural networks specialised for answer retrieval.

[7]`https://github.com/PrithivirajDamodaran/Gramformer`

[8]A question is said to be similar to another when its similarity score is above a given threshold s. Similarity scores are calculated like pertinence scores.

If Hypothesis 6 is true, then the Intelligent Explanation Generator will be able to produce better and more satisfying explanatory overviews (than the baseline YAI4Hu). This is because it will be able to anticipate, in a sense, a lot of implicit questions the user may have.

Not all words, though, require an overview. That is because, in practice, only a tiny fraction of the words in a text are helpful to explain. Indeed, many words have common-sense meanings (e.g., the words: "*and*", "*first*", "*figure*") and, therefore, should not be explained, as discussed also in Section 6.4.

To intelligently avoid jotting down unnecessary words, YAI4Edu comprises an intelligent annotation mechanism that annotates only those concepts and words that can be explained by (the knowledge graph extracted from) the textbook and other supplementary texts. More specifically, to understand whether a word should be annotated, the algorithm executes the following instructions:

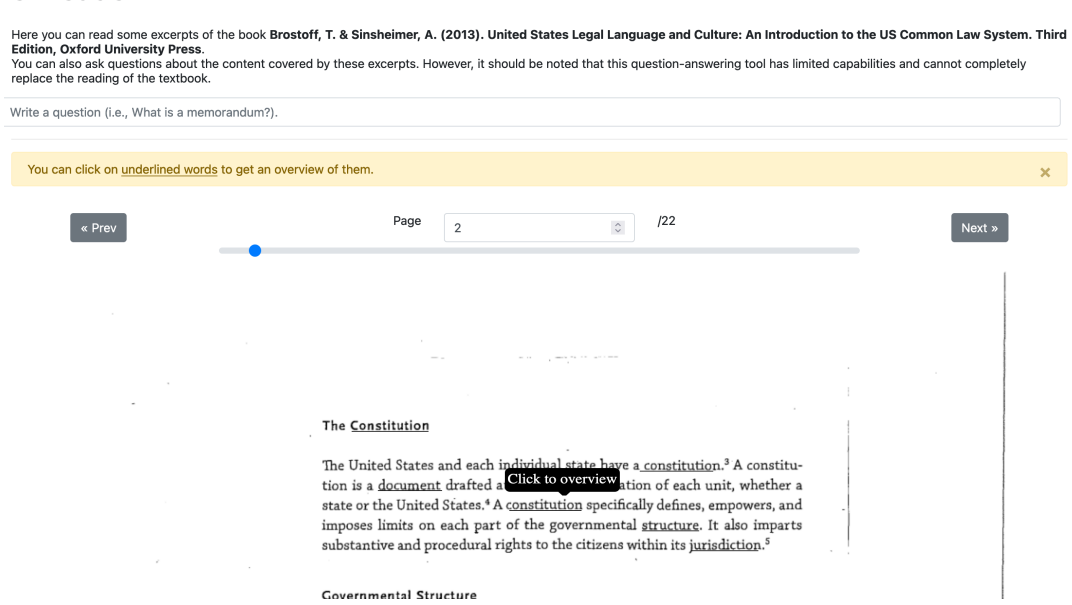
- It checks whether the word is a stop word (i.e., a commonly used word such as "and" or "or"). If so, the word is not annotated.
- If the word is not a stop word, the algorithm generates its overview through the Intelligent Explanation Generator. Then, it computes the cumulative pertinence score of the answers composing the overview; if the score is greater than a given threshold, it annotates the word.

This annotation mechanism is intended to remove noisy annotations and distractors so that the reader can focus only on the most central and well-explained concepts. Moreover, the cumulative pertinence score, which is used to understand whether a word should be annotated, can also be used as an alternative to DoX to find out which topics are best explained in the corpus of documents.

10.2 Case Study: A Textbook for Teaching How to Write Legal Memoranda

To showcase and evaluate YAI4Edu, we considered a case study in the intersection between AI and law. In particular, we applied YAI4Edu to the following **material explaining**, among other things, **how to write a legal memorandum** in a U.S. legal context for a veteran's PTSD disability claim:

Figure 10.4: ***Landing page of YAI4Edu on the case study.*** *This figure contains a screenshot of the annotated textbook [35] and the input for open-ended questioning. Clicking on underlined words opens an explanatory overview, an example of which is shown in Figure 10.1.*



- **22 pages excerpted from the textbook** "*United States Legal Language and Culture: An Introduction to the U.S. Common Law System*" [35, pp. 47-60, 93-96, 101-103, 131-132].[9]
- **5,407 open access web pages** about concepts related to the U.S. legal system coming from the encyclopaedia of the *Legal Information Institute of the Cornell Law School*[10] (5,406 web pages) and Wikipedia[11] (1 web page).
- **11,198 legal cases** on PTSD disability claims taken from the official website of the Board of Veterans' Appeals (BVA).[12]

[9] We received explicit consent from the copyright holder to use excerpts of this textbook for our experiments and the related scientific publications.
[10] `https://www.law.cornell.edu`
[11] `https://en.wikipedia.org/wiki/Law_of_the_United_States`
[12] `https://search.usa.gov/search?affiliate=bvadecisions&sort_by=&query=PTSD&commit=Search`

Altogether, the included material, comprising more than 16,000 documents, complements the primary teaching material on which YAI4Edu focuses, i.e., the excerpts of the selected textbook. In particular, the textbook is used in "*Applied Legal Analytics and AI*"[13], an interdisciplinary course at the University of Pittsburgh, co-taught by instructors from the University of Pittsburgh School of Law and Carnegie Mellon University's Language Technologies Institute. It provides "a hands-on practical introduction to the fields of artificial intelligence, machine learning and natural language processing as they are being applied to support the work of legal professionals, researchers, and administrators."

Teaching how to write a legal memorandum for the U.S. legal system is a course objective, in part, because in a *common law* system, such as the American one, the use of AI assists practitioners in efficiently retrieving legal cases for constructing arguments [154]. A *legal memorandum* is an organised document that summarises relevant laws to support a conclusion on a particular legal issue. Writing it can require legal practitioners to navigate through large databases of cases, i.e., to retrieve the definitions of technical and specific concepts or to understand which argumentation patterns are most common in a specific context. Indeed, some of the **distinguishing features of legal writing** are:

- **Authority.** The writer must back up assertions and statements with citations of authority (i.e., precedents and other decided cases).
- **Argument re-use.** A more effective memorandum may reuse existing documents as templates or argumentation patterns.
- **Formality.** The written legal document should be properly formatted according to existing standards.

Hence, legal practitioners may now be required to learn how to efficiently and effectively interact with existing AI-based technological solutions for information retrieval to speed up legal writing and to learn the complexities of legal writing. Given the task's complexity and the course's goals, we envisaged that it might be of utmost relevance and utility to design and create a tool such as YAI4Edu that could ease the acquisition of the necessary knowledge for a student to learn legal writing.

Specifically, YAI4Edu should help students understand, from real examples, how to write a legal memorandum comprising legal arguments to

[13] `https://www.law.pitt.edu/academics/courses/catalog/5719`

defend or attack a claim. In particular, students have to understand how to use statues, read and summarise cases, synthesise cases, draft a legal memorandum, and use legal concepts in writing. This may involve learning legal concepts and skills of making arguments with concrete cases selected using legal information retrieval tools, as is typical in the U.S. legal context.

Table 10.1: ***Useful statistics about the case study.*** *Column "Documents" indicates the number of documents. "Extracted Questions" provides the number of different questions extracted by the algorithm described in Section 9.3. "Concepts" is the number of different concepts/topics identified in the collection of documents. "YAI Concepts" provides the number of topics that can be explained by the algorithm described in Section 10.1. "KG Size" indicates the number of RDF triplets composing the knowledge graph (KG) extracted by the algorithm described in Section 6.2. "Tokens" shows the total number of tokens (e.g., words) in the collection of documents. "Tokens per Doc" shows the mean number of tokens per document.*

	Documents	Extracted Questions	Concepts	YAI Concepts	KG Size	Tokens	Tokens per Doc
Textbook & Web Pages	5,408	246,747	115,110	3,407	2,059,145	707,317	130.79
+ Legal Cases	11,198	1,062,716	1,410,694	4,579	52,987,778	28,630,575	2,556.75
= **Total**	16,606	1,309,463	1,525,804	7,986	55,046,923	29,337,892	1,766.7

Applying YAI4Edu to the collection of documents mentioned above, we extracted a knowledge graph of 52,987,778 RDF triplets from the BVA cases and a knowledge graph of 2,059,145 RDF triplets from the textbook excerpts and the other web pages; other statistics are shown in Table 10.1. Thanks to the property of *compositionality* of RDF graphs (introduced in Section 6.2), we also manually added a few (in the order of 10) RDF triplets to integrate missing knowledge such as the fact that the word "memo" is a synonym of "memorandum". We believe that this property of compositionality of the knowledge graph used by YAI4Edu is of utmost importance; it enables manually correcting any error produced during the graph extraction and easily integrating it with additional knowledge.

This knowledge graph helped to build an interactive and intelligent version of the textbook, as described in Section 10.1 and shown in Figure 10.2, where an input box for *open-ended questioning* and annotated (i.e., underlined) words for *overviewing* (shown in Figures 10.4 and 10.1) pro-

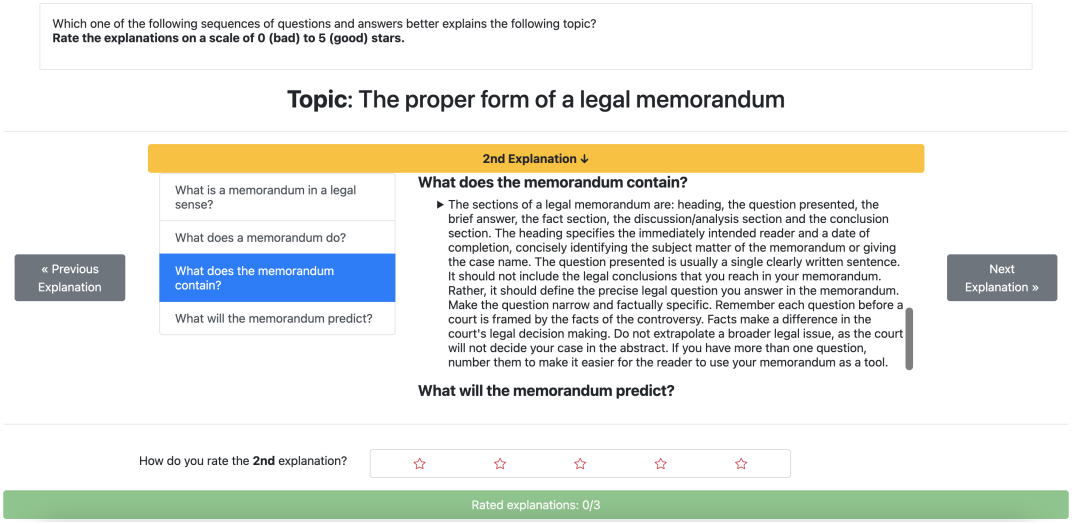


Figure 10.5: ***Screenshot of the web application used during the experiment.*** *This figure shows what the participants in the user study see during the experiment.*

vide the user with interactive elements to obtain intelligent explanations without breaking the structure of the textbook.

The choice of hyper-parameters (described in Section 10.1) is focused on generating concise and compact explanations. In particular, the hyper-parameters chosen for this instance of YAI4Edu are the following: *i)* $k = 10$ snippets with the highest DoX considered; *ii)* answer pertinence threshold $p = .57$; *iii)* maximum overview question length $L = 50$; *iv)* question similarity threshold $s = .95$; *v)* minimum and maximum answer length equal to 150 and 1000, respectively; *vi)* maximum number of questions per overview $q = 4$; *vii)* maximum number of answers per overview question $a = 2$.

10.3 Evaluation of YAI4Edu with Students

Explanations and explanatory tools may be complex artefacts whose quality depends on a wide range of different factors [195], including:

- The quality of the explainable information;
- The presentation logic with which the explainable information is used

to explain;

- The quality of the interface.

With this experiment, we are interested in *evaluating the presentation logic* used by YAI4Edu for selecting and reorganising questions and answers into explanations.

In Chapter 7, we already showed, with several examples and experiments, that a user-centred YAI is better than one-size-fits-all and static explanations. Instead of evaluating the interactive e-book with a rather time-consuming and repetitive test, we decided to focus on evaluating the one feature of YAI4Edu that should be responsible for improving the explanatory power of the e-book: the Intelligent Explanation Generator. Indeed, according to theory (cf. Chapter 3), what makes a YAI good at explaining is its ability to identify implicit and relevant questions to answer, i.e., its illocutionary force. Therefore, we devised an experiment where we directly asked real students to rate explanations for how well they adequately explain a given topic (i.e., explanandum aspect), as shown in Figure 10.5. We did it to understand the extent to which the explanations generated by our Intelligent Explanation Generator are satisfactory and whether they are better than baseline explanatory strategies (cf. Hypothesis 6).

The experiment consists of a 10-minute *within-subjects* user study where the explanations generated by two baseline explainers and YAI4Edu are evaluated by English-speaking students collected with Prolific[14], an online platform that helps recruit paid participants for online research. Participants are required to be fluent in English, be resident in English-speaking countries (i.e., USA, UK, Ireland, Australia, Canada, New Zealand, South Africa), use a device with a large screen (e.g., a laptop, a desktop computer, a tablet in landscape mode), be at least 18 years old, possess a student status and a minimum approval rating of 75% on Prolific.

The two baselines against which the Intelligent Explanation Generator is compared are slight variations. They use the same sequence of steps of the Intelligent Explanation Generator to generate their explanations, apart from the step responsible for selecting the explanatory questions, which is different. These **two baselines** are:

- An explainer that uses **randomly chosen questions** to organise the contents of an *overview*. This explainer randomly selects $q = 4$ questions, setting the maximum question length to $L = \infty$ and using a

[14] `https://www.prolific.co`

lower answer pertinence threshold $p = .3$ (and not $p = .57$ as in the Intelligent Explanation Generator). This prevents the number of questions from diminishing too much due to not finding sufficiently relevant answers.

- The generic overview generator of YAI4Hu, that uses **pre-defined and very generic archetypal questions** instead, always using the same four questions (i.e., `what` is it, `how` is it, `where` is it and `why`).

The participants were shown the explanations generated by all three explainers in a randomized order to prevent biases that may have been caused by the order of presentation. The explanations generated by the random explainer were the same for all participants, thanks to a predefined random seed. The participants evaluated each explanation on a scale of 0 (bad) to 5 (good) stars, with the specific question (that the participants were asked when rating the explanations) being, "Which one of the following sequences of questions and answers better explains the following topic?", as shown in Figure 10.5. The participants were also asked to anonymously provide the following information: age; gender; experience in legal writing; their proficiency in written English, on a scale of A1 (very low) to C2 (very high); their experience in legal writing (from *none* to *high*); how they would rate their knowledge of the U.S. legal system (on a scale of 0 (bad) to 5 (good) stars). Qualitative feedback was also solicited at the end of the test.

Table 10.2: ***Questions used during the experiment.*** *This table shows the questions extracted from the three explainers of the experiment. Specifically, "intelligent" stands for the Intelligent Explanation Generator, "generic" is the YA4Hu explainer, and "random" is the explainer that uses random questions. Note that an explainer uses fewer than four questions (the maximum) for its explanations whenever it does not find four questions with relevant answers in the knowledge graph. The sum of the relevance scores of all answers that make up each explanation is reported in the column "Cumulative Relevance".*

Topic	Explainer	Cumulative Pertinence	Questions
The proper form of a legal memorandum	generic	2,81	What
			How
			Why
	random	3,79	What is the result of a memorandum to a partner in the same firm?
			In what manner is a memorandum of points and authority usually mandatory?
			What does the memorandum usually include?
	intelligent	4,61	What is a memorandum in a legal sense?
			What does a memorandum do?
			What does the memorandum contain?
			What will the memorandum predict?
The effects of a disability	generic	3,47	What
			How
			Why
	random	4,92	What is the reason schools must determine if they have a covered disability under the Act and if that disability is severe enough?
			What can a partial disability be?
			What is an example of state statutes relating to disability retirement?
			In what manner can a partial disability be permanent?
	intelligent	5,65	What is Disability Law?
			What is disability?
			What is the result of disability in a legal sense?
			What does the disability prevent?
The elements of the legal standard a veteran needs to satisfy for a PTSD disability claim	generic	4,58	What
			How
			Why
			Where
	random	4,09	Why is element two met?
			Who is the first element of a service connection?
			Who found that the Veteran has met the first two elements of service connection?
	intelligent	5,21	What is the first element of a service connection?
			What are the elements of service connection?
			What are elements of the legal standard a veteran needs to satisfy for a PTSD disability claim?
			What is an element of appeal?

Given the case study at hand (cf. Section 10.2), the main objective of the explanatory contents is to explain how to write a legal memorandum appropriate for the U.S. legal system and a veteran's PTSD disability claim. The excerpts of the considered textbook are about legal writing, while the collection of legal cases of the BVA are about PTSD disability claims. Thus, we can say that some of the goals of the YAI for this case study are to explain: *i)* what is the proper form of a legal memorandum; *ii)* what sections should be included in a legal memorandum; *iii)* what legal standard does a veteran need to satisfy for a disability claim; *iv)* what are the elements of the legal standard a veteran needs to satisfy for a disability claim; *v)* what issues do the required elements of a disability claim raise; and *vi)* what kinds of legal arguments are appropriate for resolving such issues.

Considering that we need an experiment lasting a maximum of 10 minutes (in order to minimise costs: each participant cannot be paid less than 6£ per hour on Prolific), we chose the following **3 topics** (i.e., explanandum aspects) for evaluating the explainers:

- **Topic 1**: The proper form of a legal memorandum.
- **Topic 2**: The effects of a disability.
- **Topic 3**: The elements of the legal standard a veteran needs to satisfy for a PTSD disability claim.

The explanations for the first two topics are extracted from the textbook and web pages (the first is better explained by the textbook, the second by the web pages). Instead, the explanations for the third topic are extracted from legal cases. For more details about the explanations used in the experiment, see Table 10.2.

10.4 Discussion: Results and Limitations

We gathered 130 participants via Prolific, all of whom were students aged between 19 and 38. Each participant was paid £1. However, 28 participants had to be discarded for the following reasons:

- 26 participants were discarded because they were too quick (i.e., they spent less than 4 minutes completing the evaluation of all topics) or they skipped at least one topic (i.e., they spent less than 35 seconds on it without being a legal expert);

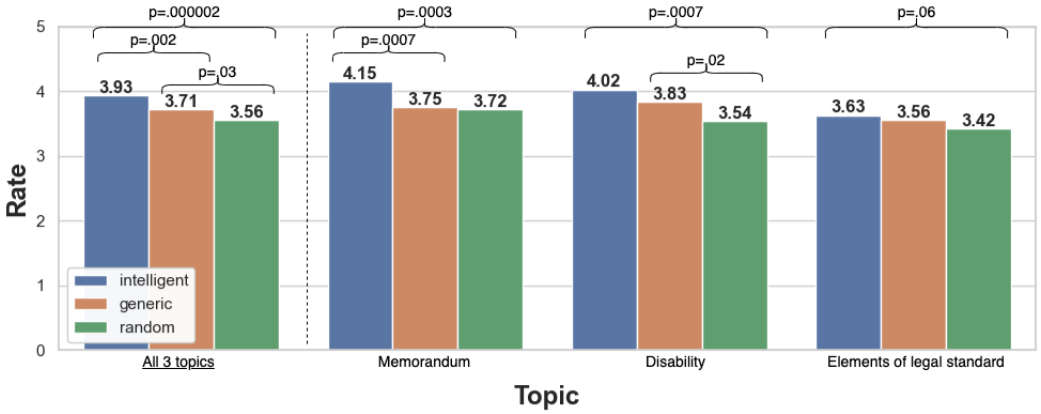


Figure 10.6: ***Experiment results.*** *This figure contains bar charts showing the average scores and interesting p-values (indicated above the curly brackets) for each topic and explainer. The bar plot labelled as "memorandum" refers to the first topic, "disability" to the second topic and "elements of legal standard" to the third topic. On the left side of the figure, one can see the results obtained by aggregating the scores for all three topics.*

- 2 participants were rejected because they reported poor knowledge of written English (i.e., A1 or A2) or wrote non-grammatical qualitative feedback.

Eventually, 102 valid submissions were collected. Of these 102 participants:

- 46 were males, 52 females, 1 identified itself as other/non-binary, and three preferred not to say;
- 81% stated that their written English proficiency level is C1 or C2, while 16% indicated a B2 level;
- 86% rated their knowledge of the U.S. legal system with a score lower than or equal to 3;
- 86 said they are not legal experts;
- 68 reported that they have low or no experience with legal writing, 29 wrote they have some experience and five that their experience with legal writing is high.

The results (shown in Figure 10.6) indicate that the *intelligent* explainer received the highest rates, followed by the *generic* one; the worst was the *random* explainer. To further validate the results and verify that the improvements of the *intelligent* explainer over the baselines are statistically significant, we performed a few one-sided Mann-Whitney U-tests (MW; a non-parametric version of the t-test for independent samples) summarised in Figure 10.6. Results clearly show that, assuming $p < .05$ as enough for asserting statistical significance, the *intelligent* explainer is superior to the baselines in terms of perceived explanatory power in all three chosen topics.

Interestingly, looking at the topics separately, we also have statistical evidence showing that the *intelligent* explainer is better than the *generic* and the *random* explainer for the first two topics, but not enough for the last one. This may be because the variance of the cumulative pertinence of the explanations about the third topic (see Table 10.2) is too low. Alternatively (and more likely), this may also be because the explanations about the last topic were extracted from a corpus of legal cases rather than textbooks or other educational contents as the other two, thus being harder to explain. In particular, this intuition is corroborated by the statistics of Table 10.1, where one can see that the legal cases have a ratio of explained concepts close to 0.32%. In contrast, the textbook and web pages have 2.96% (10 times greater). This difference in explainability between the two documents corpora may impact the quality of extracted explanations. Indeed, as pointed out by some qualitative feedback, the explanations extracted from the BVA cases contain too much (unexplained) technical jargon and too long sentences (e.g., "the first topic was easy to understand also the second one, the problem with the last one is that it was too long and was not straight to the point.")

Considering that we are performing multiple comparisons with MW, the chances of having a false comparison increase. Some statistical tools that are used in this case to reduce the chance of a type I error (false positive) are: the Bonferroni correction, the Holm–Bonferroni method, or the Dunn–Šidák correction. These tools, however, are known to increase false negatives [7]. Regardless, if we would use a Dunn–Šidák correction to adjust for 3 multiple comparisons per topic, then the minimum p-value for claiming a statistically significant result would not be $.05$ but instead something close to $.017$.

Nevertheless, the collected findings support our hypothesis, showing

that the most useful implicit questions a user may have about a collection of texts are likely to be those best answered by the whole collection. Furthermore, even if the *random* explanations have a cumulative pertinence score greater than *generic* explanations (at least for the first two topics, as shown in Table 10.2), they are evaluated as worse explanations nonetheless. This evidence further supports Hypothesis 6 (and indirectly also Hypothesis 1; cf. Section 3.2). It shows that too specific archetypal questions may be less effective than generic ones at explaining and that intelligently balancing between *generality* and *specificity* is needed, as also suggested by qualitative feedback:

- "All of the [random] explanations were ok but improvement is needed. They lack a sense of direction. Its like they go round mountains to prove one single point. All [generic] explanations were very easy to decode and they were straight to the point. All [intelligent] explanations were a mixture of first and second explanations."

- "The [random explanations] proved to be unsatisfactory for all three topics: the explanations do not follow a logical order, are incomplete and often contain incorrect or irrelevant elements. The [generic and intelligent explanations] are quite complete. The [generic explanations] seem to fit more practical questions, while [intelligent explanations] fit more theoretical ones. In my opinion, [intelligent explanations] are preferable for the topic at hand."

However, it is worth noting that our evaluation method has certain limitations. The assessment was primarily based on the students' perceptions of the quality of the explanations provided rather than measuring the actual usefulness and impact of the explanations on the students' mental models. While understanding the scale of such measurements and their implementation can be challenging, incorporating these aspects would have provided a more comprehensive evaluation of the effectiveness of our explainers.

In future work, a more in-depth analysis could be conducted to measure the actual impact of the explanations on the students' understanding, perhaps by testing their knowledge before and after exposure to the different explanations. This would help to determine the true effectiveness of the explanations in updating the readers' mental models, thus providing further insights into the performance of the explainers.

Despite these limitations, our current evaluation offers valuable insights into the students' perceptions of the explanations, which can be used to

guide further improvements and refinements of our approach.

Additionally, we collected qualitative feedback from the participants. Although optional, 81% of the participants provided feedback. Among them, 19 left positive comments (e.g., "the explanations were superb and of good quality") without suggesting any improvement or explaining their ratings, while the remaining 64 users offered suggestions. We identified **six major areas for improvement**:

- **Avoid long and redundant explanations**: suggested by 32 participants;
- **Avoid or explain legal jargon**: 24;
- **Avoid generic or incomplete information**: 18;
- **Use simpler questions**: 9;
- **Provide examples when explaining**: 7;
- **Provide better organised and compartmentalised contents**: 5.

On the whole, the comments suggest that the subjects were thoughtful. The complaints are primarily about too-long explanations, unexplained legal jargon, or generic/incomplete information. Some qualitative feedback comments ask for more conciseness, and others for less. Some participants preferred *generic* explanations over *intelligent* ones. Interestingly, one could turn this into a feature if the system could offer users a choice of generic or intelligent explanations.

Some of the most interesting feedback examples are the following:

- "The more lengthy explanations offer more details and give the reader a greater understanding but can feel a bit harder to read rather than the [generic explanations]."
- "Dividing the topic into sections is good as long as the [questions] are relevant and make sense. Relatively simple explanations supported by evidence is better in my opinion."
- "The longer explanations had more detail and were more understandable. The shorter definitions were also understandable and compact, but law should be detailed. "

- "I particularly liked the layout that included 'what, why, how', as it made the explanations easy to follow. [...] The headings that had long sentences, lost me before I began and I found it hard to decipher the explanations."
- "I would make [explanations] shorter."
- "The answers should be less vague and focus more into details."
- "The explanations with very long or involved subheadings were difficult to follow and often when there are large blocks of text, my mind tends to get overwhelmed - simply making shorter subheadings and adding more paragraph breaks to the explanations helps with this."
- "The answers for the questions could have been more concise for instance in the first topic on what a memorandum is, though I found [intelligent] and [generic explanations] to be similar in their verbiage. I found [generic explanations] always easier to understand across the board, because of it's simplistic presentation i did not spend time focusing on unnecessary details"
- "The explanations are a bit difficult to follow as they are long so as a reader you get lost in the middle of the explanation and forget what you just read on top."
- "They were pretty straightforward and easy to understand, especially because descriptions relating to the law or topics that are difficult to understand are always filled with difficult jargon, but this simplified version made it easy to understand."
- "Generally the explanations were full, but a bit difficult to digest. There were cases in which not enough was explained to fully understand what a specific legal term meant and encompassed. The explanations that had a short and quick explanation of some legal terms followed by a more prolonged and detailed explanation were for me the easiest to grasp."

While the qualitative feedback was useful in identifying potential improvements and limitations of YAI4Edu and the baselines, it is important to be cautious about taking it entirely at face value. For example, the complaint that explanations are "too long" may be a reflection of participants'

reluctance to exert mental energy, rather than an inherent problem. Balancing sufficient detail with brevity can be a challenging task. Nevertheless, the feedback does offer valuable insights for future work.

Specifically, we believe that YAI4Edu's *smart annotation* mechanism has the potential to address the jargon issue, as it can provide clear explanations for technical terms. However, we could not verify this with the experiment because it was set to last strictly 10 minutes, so intelligent annotations were excluded.

As for future work, we plan to conduct the experiment described in Section 10.3 on different textbooks or other educational materials related to law and computer science. Additionally, in response to the feedback from the user study participants, we will work on enhancing the explanation generation pipeline. We will explore the use of algorithms for intelligent summarization, paraphrasing, and text abstraction to strike a balance between brevity and detail, while reducing technical jargon.

Part III

Explanation Strategies for Reinforcement Learning Agents

Summary

In THIS part of the thesis, we discuss how the theory of explanations presented in the previous chapters can improve the learning capabilities of current state-of-the-art artificial intelligence. In particular, we focus on a specific type of artificial intelligence called Reinforcement Learning (RL), which learns from experience to make optimal sequences of decisions. We show how the SAGE-ARS model can reduce the time steps required for an RL agent to achieve a given goal optimally, thus being more *sample-efficient*. We do it starting from the hypothesis that it is possible to significantly increase the sample efficiency of RL agents by considering the space of all experiences as a particular type of explanatory space in which to apply the ARS heuristics. This is possible without changing the agent's loss function or the underlying problem's definition.

First, in Chapter 11, we provide the required background to properly understand the RL technology and our contributions. Then, in Chapter 12, we present an implementation of the SAGE-ARS model for single-agent RL, called Explanation-Aware Experience Replay (XAER), testing it on different environments and with different reward functions and algorithms (i.e., DQN, TD3 and SAC). Immediately afterwards, in Chapter 13, we discuss Dimensionality-invariant Explanatory Experience Replay (DEER), an extension of XAER for Multi-Agent Reinforcement Learning (MARL)

The content of Part III is a reworking and extension of the following article by the same author of this thesis: [201].

designed to handle better the problem of non-stationarity induced by experience replay in MARL. To show the effectiveness of DEER, we test it on typical MARL problems (i.e., multi-agent pathfinding and decentralised task assignment) and with different RL algorithms (i.e., DQN and SAC).

CHAPTER 11

Technological Background: Reinforcement Learning Algorithms, Explanations and Experience Replay

Reinforcement Learning is one of the three major paradigms of machine learning, alongside supervised and unsupervised learning. Through its ability to self-adapt and make decisions in dynamic environments, RL has been applied in various contexts, such as video games, healthcare, recommendation systems, natural language generation, autonomous driving and robotics [118]. For instance, it underpins technologies such as ChatGPT[1] and Google Ads[2].

An RL problem is typically formalised as a Markov Decision Process (MDP): a special type of stochastic sequential decision-making process which assumes that the optimality of action only depends on the current

[1] https://openai.com/blog/chatgpt/
[2] https://ads.google.com/home/

state of the world[3] [207]. In this setting, an agent interacts at discrete time steps with an external environment. At each time step t, the agent observes a state s_t and chooses an action a_t according to some policy π, a mapping (e.g., a probability distribution) from states to actions. As a result of its action, the agent obtains a reward r_t, and the environment passes to a new state s_{t+1}. The tuple $e_t = \langle s_t, a_t, r_t, s_{t+1} \rangle$ is called *state transition* or *experience*. The process is then iterated until a terminal state is reached. The future cumulative reward $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ is the total accumulated reward from time 0 to time t. $\gamma \in [0, 1]$ is the *discount factor*, representing the difference in importance between present and future rewards. The agent's goal is to maximise the expected cumulative reward (also called *cumulative return*) starting from an initial state s_t.

When there are multiple agents trained to interact (optimally) with each other (e.g., cooperating, competing), we have Multi-Agent Reinforcement Learning. In single-agent reinforcement learning scenarios, the environment changes only as a result of the actions of one agent. Instead, in MARL scenarios, the environment is subject to the actions of all agents.

In the following sections, we will review the major RL paradigms and discuss the problem of *sample efficiency*. We will also elaborate on what explanations usually are in RL, elucidating why a technique called Prioritised Experience Replay is akin to our interpretation of explaining as a user-centred process. Immediately afterwards, we will introduce MARL and some of the most common problems in MARL.

11.1 Reinforcement Learning Paradigms and the Problem of Sample Efficiency

Two major approaches to RL are value-based and actor-critic algorithms. Value-based algorithms, such as DQN [140, Deep Q-Networks], learn an optimal policy indirectly after an optimal value function is learned. A *value function* is a function that estimates the cumulative reward that an agent can obtain from a given state. In particular, DQN learns an *action-value* function $Q^{\pi}(s, a) = E^{\pi}[R_t | s = s_t, a = a_t]$[4] (also called Q-function), which estimates the expected return for selecting action a_t in state s_t and prosecuting with strategy π. Given a state s and an action a, the optimal action-

[3] This is called Markov assumption.

[4] E here means expected value. The expected value of a discrete random variable is the probability-weighted average of all possible values.

value function $Q^*(s,a) = \max_\pi Q^\pi(s,a)$ is the best possible action-value achievable by any policy. In contrast, actor-critic methods, such as SAC [84, Soft Actor-Critic], directly learn a policy function π together with a *state-value function*, also called V-function. The value of a state s reached with a policy π is $V^\pi(s) = E^\pi[R_t|s = s_t]$ and the optimal V-function is $V^*(s) = \max_\pi V^\pi(s)$.

The Q-function and the V-function can be learned by suitable function approximators, e.g., neural networks. We shall use the notation $Q(s,a;\theta)$ to denote an approximated Q-function with the parameters θ of a neural network. In DQN, we try to approximate the optimal action-value function $Q^*(s,a) \approx Q(s,a;\theta)$ using the Bellman equation [207] by learning the parameters via backpropagation. This is done through experience replay, where old state transitions are sampled from a buffer of experience populated during training. Experience is typically sampled uniformly from the buffer and is used to form training batches and thus train the neural network to estimate expected values. Therefore, Q-learning is an *off-policy* reinforcement learning algorithm: it can learn from actions taken according to a different policy. The main drawback of this method is that a reward only directly affects the value of the state action pair $< s,a >$ that led to the reward. The values of other state-action pairs are affected only indirectly through the updated value $Q(s,a)$; backpropagation to relevant previous states and actions may require several updates, slowing down the learning process and making it less efficient.

In contrast to value-based methods, actor-critic algorithms such as TD3 [77, Twin-Delayed DDPG] and SAC also parametrise the policy $\pi(a|s;\theta)$ and update the parameters θ by gradient ascent on $E[R_t]$. One consequence of this is that they can work on continuous action spaces. In particular, the TD3 algorithm is an extension of DQN that only works with continuous actions. Similarly to DQN, TD3 also learns a Q-function. In DQN, the optimal action is taken by taking *argmax* on the Q-values of all actions. In TD3, the actor is a policy network that directly produces the action, bypassing *argmax*. The policy is deterministic since it directly outputs the action. In order to promote the exploration of new actions and states, TD3 adds some Gaussian noise to the action determined by the policy.

Differently, SAC learns both a V-function and a Q-function, overcoming the problem of being limited by a fixed distribution by allowing the agent to learn the distribution with which to sample actions. This is done

through *entropy*[5] *maximisation*, which allows the agent to explore more different strategies. In particular, SAC apprehends a unimodal Gaussian policy via the reparametrisation trick [132], optimising for entropy maximisation instead of exploring using a fixed stochastic process (as the others[6]). In other words, a SAC actor aims to maximise expected reward while also maximising entropy, i.e., to succeed at the task while acting as randomly as possible. Both TD3 and SAC are off-policy as DQN.

Off-policy RL algorithms are amongst those which can better exploit experience thanks to a replay mechanism which allows revisiting past state transitions. Regardless, one of the main problems of RL is learning with little examples quickly, also called sample efficiency. *Sample efficiency* is the ratio of cumulative rewards to the number of time-steps required to train the agent, i.e., the ratio of effectiveness to efficiency. Sample efficiency indicates an algorithm that best uses the given experience samples.

Reinforcement Learning systems usually require considerable time and experience to reach average human performance. This is usually way more time than humans need. For example, DeepMind's AlphaGoZero had to play five million Go games before achieving super-human performance. In particular, emerging applications of RL require the design of *sampling-efficient solutions* to cope with the explosive growth in the dimensionality of problems. The space of states and actions to be sampled can be enormous, and without effective *experience replay* strategies, it could be unfeasible to train agents in a small amount of time [121]. In this sense, explanations can be a medium for RL agents to improve their sample efficiency.

11.2 Explanations in Reinforcement Learning

The most important field studying explanations in AI and RL is Explainable Artificial Intelligence (cf. Section 3.1). In the numerous surveys on XAI, a typical dimension used to classify explanations is the representation mode used to convey them. Within this domain, explanations are commonly conveyed via textual/visual descriptive representations of the decision criteria

[5] *Entropy* is a quantity that generally indicates how random a stochastic variable is. If a coin is weighted in such a way that it almost always comes up heads, it has a low entropy; if it is weighted evenly and has half the chance of getting either result, it has a high entropy.

[6] For example, DQN, DDPG [185] and TD3 use respectively epsilon-greedy exploration, Ornstein-Uhlenbeck noise and uncorrelated zero-mean Gaussian noise.

(i.e., *rule-based*) or with similar examples (i.e., *case-based*), as also discussed in Section 3.1. An example of a rule-based explanation is "*you will get a penalty for reaching 75, which is above the speed limit of 50*", based on the rule "*if speed is above 50, you will get a penalty*". While an example of a case-based explanation is "*you get a penalty because you are in a situation similar to this other vehicle that reached speed 74 and was previously penalised*".

Dietterich and Flann [64] frame explanation-based RL as a case-based explanatory process where prototypical trajectories of state transitions are used to tackle similar but unseen situations, while Chow et al. [50] implement a rule-based method, constraining the Markov Decision Process through Lyapunov functions.

Generally speaking, many rule-based methods for explaining to RL agents usually fall under the umbrella of a sub-discipline called *safe reinforcement learning* [78]. Safe RL includes techniques for encoding rules in the optimality criterion [50, 170] and incorporating such external knowledge into the action/state space [20]. While these methods do not generate explicit explanations, they insert safety rules into the learning process, implicitly explaining to the agent what *not* to do.

Alternatively, a famous example of a case-based method for explaining to RL agents is Imitation Learning [95], where demonstrations (as trajectories of state transitions generated by a human or expert algorithms) are used to train the RL agent. These are high quality cases/examples from a human expert or an expert algorithm. However, access to human expert data may not scale well to every domain, and not all problems dispose of accessible expert algorithms.

We are interested in sampling the most useful experiences to cover a particular agent's gap in knowledge. An *agent-centred* explanatory process is an iterative process that follows the agent through the learning process, selecting the most useful explanations for it at every time step. Below, we look at how experience replay techniques tackle this issue in RL.

11.3 Prioritised Experience Replay

RL algorithms can be either on-policy or off-policy. In particular, *off-policy* means that the experience used for training can be generated by any policy, not necessarily by the agent. Examples of off-policy algorithms are DQN, TD3, and SAC.

Off-policy RL rely on a technique called *experience replay* that stores the past state transitions (s_t, a_t, r_t, s_{t+1}) in an *experience buffer* and subsequently samples them for training. These transitions are pooled over many episodes into a replay memory, usually randomly sampled for a mini-batch of experiences.

Experience sampling can be improved by differentiating important transitions from unimportant ones. In Prioritised Experience Replay [180], the importance of transitions with high expected learning value is measured by the the magnitude of the absolute difference between the state-value or action-value the agent is estimating and what the true value is. This difference is called *Temporal-Difference (TD) error*. Experiences with a larger TD error are sampled more frequently, as the TD error quantifies the unexpectedness of a given transition. This prioritisation can lead to a *loss of diversity* and *introduce biases*. Bias in Prioritised Experience Replay occurs when the experience distribution is changed without control, modifying the solution to which the estimates will converge. This bias can be corrected through *importance-sampling weights*, as explained by Schaul et al. [180]. Instead, loss of diversity is mitigated with stochastic prioritisation, interpolating between pure greedy prioritisation and uniform random sampling. Sampling probability is monotonic regarding transition priorities while assuring a non-zero probability even for minimum-priority transitions.

Notably, many approaches to Prioritised Experience Replay in RL can be re-framed as mechanisms for achieving agent-centrality, re-ordering experience by relevance in the attempt of explaining to the agent and selecting the most useful experience. In particular, an *agent-centred explanatory process* is an iterative process that follows the agent through the learning process. It selects the most useful explanations for it at every time step.

Over the years, many human-inspired intuitions behind Prioritised Experience Replay drove researchers towards improved, more sophisticated and agent-centred mechanisms to RL [206, 226, 227]. Among these works, the closest to a fully agent-centred explanatory process is Experience Replay Optimisation [227], which moves towards agent-centrality by providing an external black-box mechanism (or experience sampler) for extracting arbitrary sequences of information out of a flat (i.e., no *abstraction* involved) experience buffer. The experience sampler is trained to select the most "useful" ones for the learning agent. However, due to its non-explainable nature, it is unclear whether the benefits given by Experience Replay Optimisation are due to the overhead the experience sampler gives,

increasing the number of neurons in the agent's network.

Another work trying to achieve agent-centrality in this sense is Attentive Experience Replay [206], suggesting the prioritisation of uncommon experience that is also on-distribution (related to the agent's current task). However, as the previous one, this work also falls short of explicitly organising experience in an abstract-enough way by conveying human-readable explanations to the agent. Conversely, Hierarchical Experience Replay [226] has attempted to address the abstraction issue to simplify the task to the agent, decomposing it into sub-tasks. However, Yin and Pan [226] do not do so in an agent-centred and goal-oriented way, given that the sub-task selection is uniform and not curricular. In contrast, a curricular approach for training RL agents was proposed by Ren et al. [166]. They exploited Prioritised Experience Replay and the intuition that simplicity is inversely proportional to TD errors but did not exploit any abstract and hierarchical representation of tasks.

11.4 Multi-Agent Reinforcement Learning

Multi-Agent Reinforcement Learning is a subfield of reinforcement learning. It focuses on studying the behaviour of multiple learning agents coexisting in a shared environment.

MARL agents can be trained in a centralised manner but executed in a decentralised manner (and vice versa), or they can be both trained and executed in a centralised/decentralised manner. MARL is said to have *centralised execution* when the actions taken by agents are chosen by a single process, while the *decentralised execution* paradigm is when agents are not coordinated by any centralised agent. To manage many agents, a decentralised structure in which each agent autonomously executes its policy to maximise individual performance is more scalable.

In centralised execution, a central controller observes the environment and distributes actions (and rewards) to individual agents. However, this approach becomes impractical as the number of agents increases. In particular, centralised (execution) MARL can be seen as an instance of single-agent RL, in which there is a single meta-agent that makes all decisions in a joint action space. In contrast, decentralised (execution) MARL poses the following (still open) challenges that further complicate the optimal training of an agent.

Partial observability due to the unknown state of other agents. In

a decentralised execution setting, the local decision-making system of an individual robot is inherently incomplete, since other agents' unobservable states may affect future values [79]. In this scenario, simply using recurrent neural networks[7] to keep track of past observations, as in [14], is not enough. Thus, one of the main solutions to this problem is given by the adoption of a differentiable communication channel, i.e., a Graph Neural Network (GNN)[8], for agents to deal with incomplete observations through complex multi-agent coordination. In particular, when deploying GNNs in the context of multi-robot systems, individuals are modelled as nodes, the communication links between them as edges, and the internal state of each robot as graph signals, as shown by Blumenkamp et al. [23] or by Li et al. [122].

High dimensionality due to the number of transitions per episode increasing with the number of agents. The dimension of the observation space often grows combinatorially with the number of agents, making harder for a MARL agent to learn a model of the environment and thus an optimal policy. When the observation space is too big (i.e., high-dimensional) or the amount of observations per episode are too many due to decentralisation, what happens is that relevant state transitions in a relatively small or too big buffer might not be replayed at all. One of the most famous solutions to mitigate this issue is certainly Prioritised Experience Replay [180] (cf. Section 11.3). A complementary solution to Prioritised Experience Replay is also *Global Distribution Matching* [98], a prioritisation scheme designed to preserve diversity in the experience buffer and guarantee a training distribution that matches the test distribution even with a small experience buffer. In particular, this strategy consists in randomly assigning a drop priority to every state transition inserted in the experience buffer, replacing the transitions with the lowest priority in the buffer instead of the oldest ones, in order to maintain a random sample over the global distribution even though experiences arrive sequentially and the global distribution is not known in advance.

[7] Recurrent neural networks are a class of neural networks that are naturally suited to processing time-series data and other sequential data [65].

[8] A Graph neural network is a class of artificial neural networks for processing data that can be represented as graphs [179].

Non-stationarity due to unpredictable changes in agents' policies. The action taken by one (decentralised) agent can influence the reward of other (decentralised) agents and the evolution of the environment, invalidating the stationarity hypothesis to establish the convergence of RL algorithms [228]. *Non-stationarity* means that each agent can enter a cycle of adaptation to other agents due to transitions and rewards that depend on the actions of all agents whose decision policies are constantly changing during the learning process . The problem of non-stationarity in decentralised execution is extremely complex and intrinsically unavoidable. Hence, one of the most common and naive approaches consists in employing *independent learning*, i.e., single-agent learning algorithms that intentionally ignore the effect of other strategic agents in their environment. Even though independent learning, in the most generic case, does not have any theoretical convergence guarantee [228], suffering from non-stationarity and an increased observation space (due to the fact that independent agents have independent observations), empirically it may achieve satisfiable performance, as pointed out also by [75, 228]. However, using independent learning with off-policy MARL has a further issue. In fact, the dynamics generating the state transitions in the agent's replay memory may no longer reflect the current learning dynamics, making experience replay ineffective in practice [75].

CHAPTER 12

Explaining Rule-Dense Regulations to Reinforcement Learning Agents

Human beings learn through explanations, and our ability to explain and transmit knowledge is the fuel that has propelled almost all the scientific and technological advances we have witnessed over the past millennia. Therefore, if intelligence is indeed about being able to explain, it follows that a correct understanding of the act of explaining can, in principle, be used to create more intelligent machines capable of better understanding (human) knowledge and to pass it on to (human) explainees.

We have seen in the previous chapters how the SAGE-ARS model can help build more user-centred YAI software, supporting human explainees in dealing with large explanatory spaces more effectively and satisfactorily.

The work presented in Chapter 12 was developed in collaboration with Alex Raymond from the University of Cambridge [201]. *F. Sovrano*: conceptualization, methodology, software (XAER and testing environments), data curation, original draft preparation, visualization, investigation, validation, review and editing. *A. Raymond*: conceptualization, software (testing environments of Section 12.2 only), data curation, original draft preparation, visualization, review and editing.

Empirical results indicate that a better understanding of how explanatory processes work can aid human learners in acquiring information better. Therefore, in this chapter, we discuss how the theory presented in Part I can help improve the learning capabilities of current state-of-the-art RL, in addition to producing more effective explanatory tools for people.

RL differs from supervised learning in that it doesn't require examples of desired outcomes (like an annotated dataset) or the explicit correction of sub-optimal outputs. Instead, RL relies on a well-defined reward function to guide the learning process. Designing this reward function can be challenging because it needs to effectively represent the desired outcomes and encourage appropriate behaviour. The primary focus of RL is to discover an optimal sequence of actions by striking a balance between *exploration*, which seeks to fill in missing knowledge, and *exploitation*, which leverages acquired knowledge.

Due to these characteristics, RL agents are well-suited for the iterative exploration of an explanatory space. This makes them compatible with the role of an explainee, as envisioned by the SAGE-ARS model.

Now, let us suppose we want to explain a complex rule base (e.g., a road code) to an RL agent (e.g., a self-driving car). A first naive approach could be to take the regulation as it is and feed it to the agent as part of the environment. Nevertheless, this would assume the agent has some ability to automatically *understanding* the regulation and use it, regardless of how it is represented. Another naive and task-specific approach could be to encode the rule base as a sequence of mathematical constraints for the agent's policy, integrating them in the loss function of the RL algorithm. Though this approach may not always work with every RL algorithm, it may also result in sub-optimal performance. A conventional model-free RL agent does not usually receive a representation of the rules of the system (i.e., the regulations). Instead, it learns from experience encoded into state transitions. In other words, the challenge of explaining written knowledge to RL agents is that they do not speak any natural language (e.g., English) and learn through examples rather than words and textual explanations. However, human-regulated environments often rely on legislation and complex sets of rules.

Historically, RL methods have been typically tested in environments with relatively sparse rules and exceptions [104]. Denser regulations appear in applications of RL for autonomous vehicle research, but such rule sets are often fixed in terms of complexity [120]. With large numbers of

corner cases arising as a consequence of dense rule sets, generating a sufficiently diverse set of experiences and exposing these exceptions to an RL agent can be challenging. Some works in literature propose to sample past experiences related to those exceptions, heuristically revisiting potentially important events. Among them, the technique of Prioritised Experience Replay (cf. Section 11.3) looks at over-sampling experiences that the agent's learned model most poorly captures. However, this mechanism does not necessarily focus on the cause of events or their exceptional nature.

Hereby, we pursue the intuition that explanations have the potential to boost the performance of RL agents in complex environments. This is why we draw inspiration from user-centred explanatory processes for humans and design a set of heuristics and mechanisms for Prioritised Experience Replay to explain complex regulations to a generic off-policy RL agent. A central design challenge towards this goal is integrating explanations into computational representations. Approaches such as encoding the rule set (or part of it) into the agent's observation space may incur severe re-training overhead even under minimal rule-set changes, as the semantics of the regulation are explicitly provided as input [109]. This minimises compatibility with extant methods and may need to be clarified whether differences in performance are due to changes to the architecture or the complexity of the rule set. On the contrary, we propose a solution that is agnostic to explicitly engineering state and observation spaces, using an explanation-aware experience replay mechanism.

In our approach, we avoid explicit representations of the rule-set (i.e., rule-based explanations [29]) by instead representing the meaning of the regulations *as organised collections of examples* (i.e., case-based explanations [1]). In the traditional sense, these explanations do not need to be *understood* by the agent. However, they can still convey meaning if the example is labelled/explained in a semantic and meaningful process. In a ludic example, suppose a young man called Luke is taking hyperspace flight lessons from his exasperated friend Chewbacca. However, he does not understand a single word of Shyriiwook, the tutor's language. With sufficient repetition, Luke can associate distinct Wookiee growls (and punishments) with categories of experienced episodes, even if the content of the message is in an unknown language. Eventually, Luke would learn the meaning of the most relevant utterances by associating them with the experienced consequences.

Therefore, we make the following hypothesis.

Hypothesis 7 (Experience buffers are explanatory spaces)**.** *The experience buffer of an off-policy RL algorithm is a special kind of explanatory subspace (cf. Section 5.3). So, it is possible to use the ARS heuristics (cf. Section 5.4) on an experience buffer to generate more usable explanations for an RL agent, thus improving its sample efficiency. These explanations can be seen as ordered sequences of state transitions sampled through an experience replay mechanism.*

In other words, we propose to modify conventional experience replay structures by dividing the replay buffer into several clusters/hyperedges following the ARS heuristics, where each cluster represents a distinct explanandum aspect associated with a set of experiences that serve as explanatory examples. We call this process *Explanation-Aware Experience Replay* (XAER; see Figure 12.1) and integrate this technique into three seminal learning algorithms: DQN, TD3 and SAC (cf. Section 11.1).

In summary, we state the following contributions:

- We show how distinct types and instances of explanations can be used to partition replay buffers and improve the rule coverage of sampled experiences.
- We design discrete and continuous environments (Grid Drive and Graph Drive) compatible with modular rule sets of arbitrary complexity (cultures). This leads to 9 *learning tasks* involving both environments with different levels of rule complexity and reward sparsity. These serve as a platform to evaluate how RL agents react to changes in rule sets whilst keeping a consistent state and action space.
- We introduce XAER-modified versions of traditional algorithms such as DQN, TD3, and SAC and test the performance of those modified versions in our proposed environments.

Upon experimenting on the proposed continuous and discrete environments, our key insight is that organising experiences with XAER improves the *sample efficiency* of an RL agent (compared to traditional Prioritised Experience Replay) and can be able to reach a better policy where traditional Prioritised Experience Replay may fail to learn altogether.

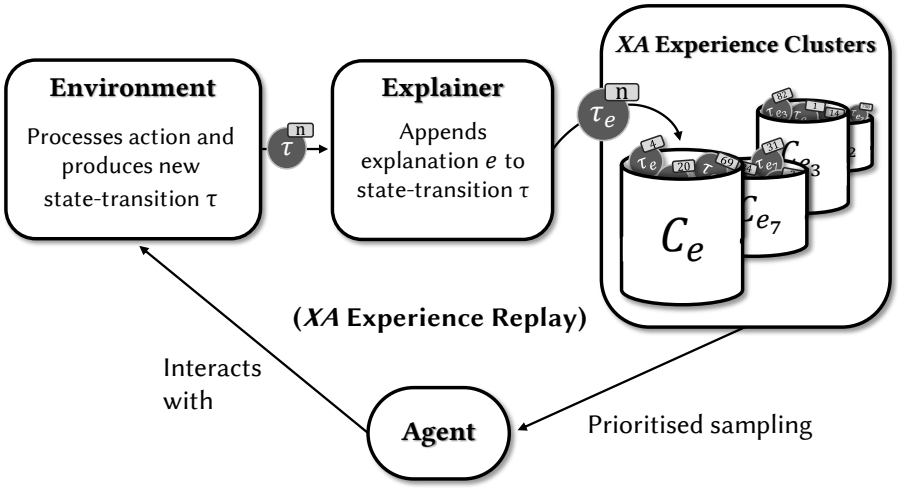


Figure 12.1: ***Overview of XAER.*** *The explainer labels a state transition τ with an explanation e, stored in a cluster (C_e) containing other experiences labelled with the same explanation.*

12.1 XAER: Explanation-Aware Experience Replay

We propose a transformation of rule-based explanations (e.g., given by a rule-set/culture) to case-based explanations (experience), which are compatible with experience replay. Drawing from an epistemic [131] interpretation of explanations, we argue that a central aspect of providing case-based explanations to an RL agent comes from meaningfully re-ordering experience to a greater degree, organising the experience buffer as an explanatory space (cf. Definition 10). The intuition behind how these case-based explanations are constructed is: *"explanations for RL agents are a simple set of relevant state transitions representing abstract-enough aspects of the problem to be solved (i.e., the explanandum)."* This intuition motivates the heuristics of *abstraction*, *relevance*, and *simplicity* (ARS, for short) described in Section 5.4.

Our use of explanations in RL is aligned to Holland's [91], and Achinstein's [2] philosophical theories of explanations (cf. Chapter 2). In fact, in the former, explaining is framed as a process of revising belief whenever a new experience challenges it. In the latter, explaining is the attempt to answer questions (such as `why`, `what`) in an agent-centred way. So,

leaning on the concept of *explanation awareness*, our heuristics facilitate information acquisition via the organisation of experience buffers.

Consider a problem where an RL agent has to learn a policy to optimally navigate through an environment with sophisticated rules and exceptions (e.g., a real traffic regulation with exceptions for particular types of vehicles). Let the state transition $\tau = (s_t, a_t, r_t, s_{t+1})$ denote the transition from state s_t to state s_{t+1} by means of action a_t, yielding a reward r_t. We assume the environment is imbued with explanatory capabilities via an *explainer*. For example, this explainer could be a function capable of capturing the run-time control flow of the reward function, providing it as explanation for a given reward. Note that the explanations generated by the explainer can have virtually any representation, be it human-understandable or not, provided they are distinct and serve the purpose of labelling different clusters.

Definition 11 (Explainer)**.** *The explainer $\epsilon : \Omega \rightarrow ES$ is a function that maps a list of state transition tuples $\tau \in \Omega$ to an explanation $e_r \in ES$, where Ω is the space of possible state transitions and ES is the explanatory space, i.e., the space of all possible explanations.*

An agent with more diverse experiences regarding the *reasons* (explanations) associated with rewards will have a better chance at converging towards a policy that better represents the underlying rule-set. Therefore, we posit that the more complex the environment is in terms of rules, the more useful for an agent is to be *explanation aware (XA)*, as it would ensure a more even distribution of experiences with regards to different reasons justifying rewards. This diversity of explanations culminates in a clustering that is semantic by nature, and transitions are partitioned according to the explanation representing its reward.

Definition 12 (XA Clusters)**.** *Let $\tau_e = (s_t, a_t, r_t, e_{r_t}, s_{t+1})$ be a XA state transition represented by the explanation e, where $\tau_e : \tau \times e_r$, $\tau \in \Omega$ and $e \in ES$. Let Ω be the set of all state transitions. We say $\mathcal{C} = \{C_{e_1}, \ldots, C_{e_k}\}$ is the set of XA clusters seen in Ω, where k is the number of different explanations seen.*

In other words, we argue that experience buffers may act as explanatory spaces for RL agents. We introduce our adaptation of the ARS heuristics to Reinforcement Learning, below.

12.1.1 Abstraction: Clustering Strategies

The purpose of the *abstraction* heuristic is to regulate the granularity of the explanations, hence of the experience clusters (or hyperedges). Our abstractions are based on the understanding that explanations are answers to questions, as discussed in Chapter 3. Hence, explanations may have different granularity defined by the level of detail of the question they answer.

More specifically, the `HOW` explanations we consider answer the question "How well is the agent performing with this reward?". This type of explanation can be produced by studying the average behaviour of an agent. For example, if an episode has a cumulative reward greater than the running mean, the explanation indicates that the agent behaves better than average. Hence, these `HOW` explanations do not need to be designed with any specific domain knowledge, as they are governed exclusively by the agent's performance. On the contrary, the `WHY` explanations we consider answer the question "Why did the agent achieve this reward?". These `WHY` explanations could depend on an explainer function with task/domain knowledge that can distinguish and cluster types of transitions (see Example 1, below). Furthermore, `WHY` and `HOW` explanations (or any other type) can be combined so that the explanation would answer both the associated questions.

In order to compose the experience buffer, represented by the set of experience clusters $\mathcal{C} = \{C_{e_1}, \ldots, C_{e_k}\}$, we consequently devise the following clustering strategies, for each explanation type:

1. `HOW`: The experience buffer is divided into 2 clusters C_{better} and C_{worse}, where C_{better} contains batches with rewards greater than the running mean of rewards, and vice-versa (given a sliding window of a defined size).

2. `WHY`: The number of clusters is equivalent to the number of distinct explanations available. Suppose a batch can be explained by multiple explanations simultaneously. In that case, we select the explanation associated with the smallest cluster (most under-represented), and the batch is associated with the corresponding cluster.[1]

3. `HOW+WHY`: a combination of `HOW` and `WHY` strategies. There are two custom C_{better} and C_{worse} clusters for every `WHY` explanation, formed after their concatenation.

[1] Since buffers will be prioritised and clusters will be fairly represented, there is no need for duplicating the batch across multiple clusters.

Example 1. *Suppose a hypothetical football environment with a* `WHY` *explainer function. This function could either be part of the environment (a logical mechanism that recognises when certain states are reached and produces a state label) or an external mechanism that receives state transitions as input and produces explanations. The explanations could be generated by the game's rules, such as "goal", "offside", or "foul". The corresponding* `WHY` *clusters would be* $\mathcal{C} = \{C_{goal}, C_{offside}, C_{foul}, \ldots\}$*, where each cluster would contain a set of state transitions associated with each label. Alternatively, clusters would be* $\mathcal{C} = \{C_{goal\_better}, C_{goal\_worse}, C_{offside\_better}, \ldots\}$*, if* `HOW+WHY` *were used.*

After clustering state transitions using the prior clustering strategies, we propose mechanisms for assessing the *relevance* of specific state transitions during learning.

12.1.2 Relevance: Intra-Cluster Prioritisation

Prioritisation mechanisms are used for organising information given their relevance to the agent's objectives.

The priority of a batch is usually estimated by computing its loss according to the agent's objective [180]. In DQN, TD3, and SAC, relevance is estimated by the agent's absolute TD error. The closer to 0, the lower the loss and the relevance. The intuition is that batches with TD error equal to zero are of no use since they represent an already solved challenge. In our method, this *relevance* heuristic can be combined with the clustering strategy mentioned above by sampling clusters in a prioritised way (by summing the priorities of all its batches) and then performing prioritised sampling of batches from the sampled cluster.

12.1.3 Simplicity: (Curricular) Inter-Cluster Prioritisation

Occam's Razor [24] suggests that, given two explanations for the same phenomenon, the simpler one should be preferred. In human explanations, *simplicity* is often used as a heuristic [101, 145]. We aim to adhere to simplicity principles by adopting a curricular learning approach, which organizes learning materials in a structured, progressive manner, and by selecting minimal, straightforward explanations.

Clustered Prioritised Experience Replay changes the real distribution of tasks through over-sampling. Assuming that the whole experience buffer has a fixed and constant size N and that the experience buffer contains $|\mathcal{C}|$ different clusters, let S_{min} and S_{max} be the minimum and maximum size of a cluster. Any new experience is added to a full buffer by removing the oldest one within buffers having more elements than S_{min}.

Suppose all the clusters have the same size (therefore $S_{\text{min}} = S_{\text{max}}$). In that case, replaying the task's cluster with the highest (TD error) priority might push the agent to tackle the exceptions before the most common tasks, preventing the agent from learning an optimal policy faster. The assumption here is that exceptional tasks (exceptions) are less frequent.

On the contrary, if $S_{\text{min}} = 0$ and $S_{\text{max}} = \infty$, the size of a cluster would depend only on the real distribution of tasks within a small sliding window, as in traditional Prioritised Experience Replay, thus preventing over-sampling. The presence of clusters helps over-sampling batches likely related to under-represented tasks and learning to tackle potentially hard cases more efficiently.

Consequently, we posit that S_{min} shall be large enough for effective over-sampling while having $S_{\text{max}} > S_{\text{min}}$ being dependent on the real distribution of tasks. This will push the agent towards tackling the most frequent and relevant tasks first, analogously to curricular learning. We define a hyper-parameter to control the *cluster size proportion*.

Definition 13 (Cluster Size Proportion)**.** *In order for all clusters to have a size $S_{min} \leq S \leq S_{max}$, we set $S_{max} = S_{min} + (\xi - 1) \cdot |\mathcal{C}| \cdot S_{min}$, where $\xi \geq 1$ represents the cluster size proportion.*

Therefore, $S_{\text{min}} = \frac{N}{|\mathcal{C}| \cdot \xi}$ can be easily controlled by modifying ξ, as shown in Figure 12.2. We enforce $S_{\text{min}} < S_{\text{max}}$ when $\xi > 1$. Consequently, for *curricular prioritisation*, if the cluster's priority is (for example) computed as the sum of the priorities of its batch, and $\xi > 1$ is not too large (e.g., $\xi = 5$), the resulting cluster's priorities will reflect the real distribution of tasks while smoothly over-sampling the most relevant tasks. This avoids over-estimation of the priority of a task. As ξ gives us control of the degree of on-policyness, different values of ξ might perform better on an algorithm and environment basis[2]. Higher values of ξ mean that the distribution of state transitions reflects more transitions seen within the current policy, thus advantageous for entropy-maximisation algorithms such

[2]However, tuning for ξ seems relatively simple, and a grid search on $\xi \in \{1, 2, 3, 4, 5, \inf\}$ might suffice for most cases.

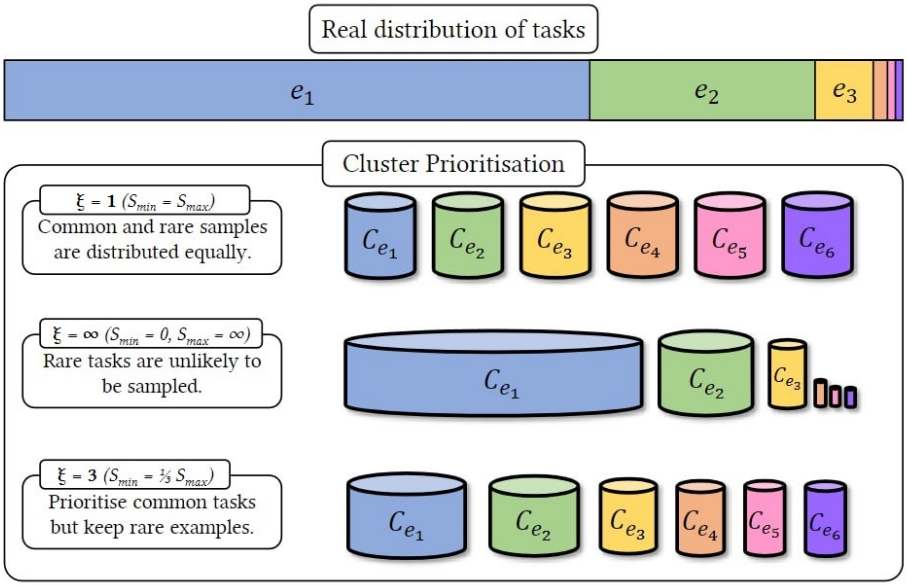


Figure 12.2: ***Overview of the simplicity heuristic.*** *This figure shows how the hyper-parameter ξ impacts cluster prioritisation and can be used to give higher priorities to simpler (i.e., less exceptional; more common) explanandum tasks (represented by clusters).*

as SAC. Likewise, fully off-policy algorithms such as DQN may exhibit superior results with low values of ξ (e.g., $\xi = 1$).

With those mechanisms in place, we propose new environments to evaluate agents' performance when subjected to complex rule sets.

12.1.4 Annealing the Bias

Similarly to Prioritised Experience Replay [180], sampling state transitions from prioritised clusters might produce unwanted bias. The standard de-biasing function of Prioritised Experience Replay weighs expected values using the normalised weight $\frac{P(\bar{\tau})}{P(\tau)} \in [0, 1]$, where $P(\tau)$ is the probability of sampling a state transition τ from the whole buffer and $\bar{\tau}$ is the state transition with the lowest probability for the whole buffer. We adopted the de-biasing function of Prioritised Experience Replay by changing the formula to consider that state transitions are sampled from clusters (which are, in turn, sampled). Therefore, the de-biasing function of XAER computes the joint probability of sampling both a cluster c and a state transition τ.

Considering that the two events are not independent, we compute this joint probability as $P(c) \cdot P(\tau|c)$. Hence, the normalised weights produced by the de-biasing function of XAER are given by $\frac{P(\bar{c} \cap \bar{\tau})}{P(c \cap \tau)}$, where $P(\bar{c} \cap \bar{\tau})$ is the lowest possible probability, considering any couple of clusters and state transitions.

12.2 Environments for Evaluating XAER

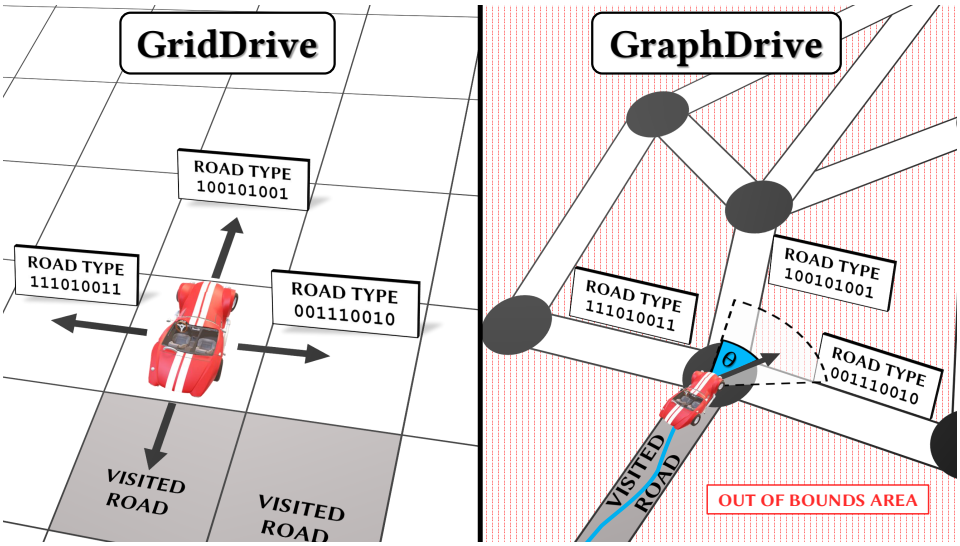


Figure 12.3: ***Diagrams representing the Grid Drive and Graph Drive environments.*** *In Grid Drive, the agent has a discrete action space and must observe the properties of neighbouring cells to make a decision compatible with the rule set, choosing one direction and a fixed speed. Graph Drive is a harder environment where the agent's action and observation spaces are continuous. In it, kinematics are considered, and the agent must learn the rules governing penalties and accelerate and steer without going off-road. Both environments aim to visit as many new roads as possible without infringing rules.*

Real-life air/sea/road traffic regulations are often complex, and their mastery is crucial to orderly navigation. Many realistic settings have several exceptions that must be considered (e.g., ambulances are not subjected to some rules in emergencies, and sailing boats have different priorities if on wind power). To evaluate the effects of XAER in a diverse configuration

space of environments, we developed modular environments that allow us to systematically change its properties in evaluation. These environments are namely:

- Grid Drive: a grid-like environment compatible with DQN, where agents can take discrete actions (e.g., move left, right, up, down).
- Graph Drive: a graph-like environment compatible with TD3 and SAC, where agents can take continuous actions (e.g., steer by 30°, accelerate by 0.5 $\frac{m}{s^2}$).

Diagrams representing them are shown in Figure 12.3.

Our environments allow agents to experience the same rules (our Easy, Medium, and Hard rule sets) in discrete and continuous state-action spaces and with frequent and sparse rewards. The agent must understand the complex regulation governing the penalty system. To implement our rule sets, we use *cultures* [162, 163]: a mechanism to encode human rule sets as machine-compatible argumentation frameworks imbued with fact-checking mechanisms. These can serve as explainer functions to produce rule-based explanations from an agent's behaviour which are then converted by XAER in case-based explanations for RL agents.

In particular, every episode involves an initialisation of the grid or graph (for Grid Drive or Graph Drive, respectively) with random roads and randomly-sampled agent properties. The RL agent is encouraged to drive for as long as possible until it either achieves a maximum number of steps or breaks a rule (terminal state). All environments are instantiated in versions with three different cultures (rule sets) according to their *levels of complexity*:

- *Easy*: 3 properties (2 for roads, 1 for agents), five distinct rules to explain.
- *Medium*: 7 properties (5 for roads, 2 for agents), 12 distinct rules to explain.
- *Hard*: 15 properties (9 for roads, 6 for agents), 20 distinct rules to explain.

12.2.1 Grid Drive: a Discrete Environment for Testing XAER on DQN

Grid Drive consists of a 15×15 grid of cells, where every cell represents a different type of road (see Figure 12.3, left), with base types (e.g., mo-

torway, school road, city) combined with other modifiers (roadworks, accidents, weather). Each vehicle will have a set of properties that define which type of vehicle they are (e.g., emergency, civilian, worker). Complex combinations of these properties will define a strict speed limit for each cell, according to the culture.

Actions. An action (d, s) consists of a direction $d \in \{N, S, E, W\}$ and a speed s where $0 < s \leq 12$.

Observations. A sample in the observation space is a tuple (o_v, o_r, M, x, y) where o_v denotes the concatenation of the vehicle's properties (including speed), o_r is the concatenation of all neighbouring roads' properties, M is a $15 \times 15 \times 2$ boolean matrix keeping track of visited cells, and (x, y) represents the vehicle's current global coordinates.

Rewards. Let $0 < s' \leq 1$ denote the *normalised* speed of the agent in that step. Rewards are given at every step, given the following criteria:

$$\begin{cases} \text{-1 (terminal)} & \text{if breaking the speed regulation} \\ 0 & \text{if on previously-visited cell} \\ s' & \text{otherwise (new cell, within the speed limit)} \end{cases}$$

Explanations. `WHY` explanations are attached to state transitions. These explanations are:

$$\begin{cases} \textit{“not visiting new roads”} & \text{if on previously-visited cell} \\ \text{the violated rules} & \text{if breaking the speed regulation} \\ \textit{“is moving”} & \text{otherwise} \end{cases}$$

12.2.2 Graph Drive: a Continuous Environment for Testing XAER on TD3 and SAC

Graph Drive consists of a Euclidean representation of a *planar* graph with n vertices and m edges (see Figure 12.3, right). The agent starts at the coordinates of one of those vertices and has to drive between vertices (called “intersections”) in continuous space with Ackermann-based non-holonomic motion. Edges represent roads and are subjected to the same rules with properties as those seen in Grid Drive, plus a few additional rules to encourage the agent to stay close to the edges. The incentive is to drive as long as possible without committing speed infractions. In this setting, the agent must learn a control input that keeps the vehicle on the road and respects speed limits and restrictions that may vary on a case-by-case basis.

We test two variations of this environment: one with dense and another with sparse rewards.

Actions. A sample (θ, a) in the action space consists of a steering angle θ where $-\frac{\pi}{4} \leq \theta \leq \frac{\pi}{4}$, and an acceleration a where $-7 \leq a \leq 1$. Acceleration and deceleration ranges are chosen given road car standards (in m/s^2) [25, 71].

Observations. A sample in the observation space for Graph Drive is a tuple (o_v, o_r, o_j), where o_v denotes the concatenation of the vehicle's properties (car features, position, speed/angle, distance to path, intersection status, number of visited intersections), o_r is the concatenation of the properties of the closest road to the agent (likely to be the one the agent is driving on), and o_j is the concatenation of the properties of roads connected to the next intersection.

Rewards (dense version). Let $0 < s \leq 1$ denote the *normalised* speed of the agent in that frame, and let n be the number of *unique* intersections visited in the episode. Rewards are assigned at every frame, given the following criteria:

$$\begin{cases} \text{-1 (terminal)} & \text{if breaking the speed regulation} \\ \text{-1 (terminal)} & \text{if off-road or U-turning outside intersection} \\ 0 & \text{if on intersection or previously-visited road} \\ s & \text{otherwise (on the road, within the speed limit)} \end{cases}$$

Rewards (sparse version). In this version, the agent will get a null (zero) reward when moving correctly. Positive rewards only appear when the agent manages to acquire a new intersection. Therefore, the agent must drive the entire road correctly to get any positive reward. Rewards are given according to the following criteria:

$$\begin{cases} \text{-1 (terminal)} & \text{if breaking the speed regulation} \\ \text{-1 (terminal)} & \text{if off-road or U-turning outside intersection} \\ 0 & \text{driving normally or on acquired intersection} \\ 1 & \text{the instant a new intersection is acquired} \end{cases}$$

Explanations. `WHY` explanations are attached to state transitions simi-

larly to Grid Drive (cf. Section 12.2.1). These explanations are:

"is off-road"	if off-road
"is U-turning"	if U-turning outside intersection
the violated rules	if breaking the speed regulation
"is on a new intersection"	if on a new intersection
"is on an old intersection"	if on an old intersection
"is moving"	otherwise

12.3 Evaluation of XAER and Results Discussion

In this section, we describe our experimental setup and present results obtained in our proposed environments with XAER versus traditional Prioritised Experience Replay. We trained three baseline agents with traditional Prioritised Experience Replay (DQN/Rainbow, SAC, and TD3). For each of the three baseline algorithms, we trained three XAER versions with different clustering strategies, using `HOW`, `WHY`, and `HOW+WHY` explanations (see Section 12.1.1).

We show results for `HOW+WHY` explanations *without* the simplicity heuristic (prioritised clustering), i.e., clusters are sampled uniformly. For a total of 12 XA agents, we call the XAER-equipped versions of DQN, SAC, and TD3 *XADQN*, *XASAC*, and *XATD3*, respectively. DQN and XADQN agents are applied to Grid Drive (discrete), whilst SAC, TD3, XASAC, and XATD3[3] were trained separately on Graph Drive with dense and sparse rewards (continuous).

The neural network adopted for all the experiments is the default one implemented in the respective baselines (although better ones can certainly be devised), and it is characterised by fully connected layers of few units (e.g., 256) followed by the output layers for actors and/or critics, depending on the algorithm's architecture. XAER methods introduce the cluster size proportion (ξ) hyper-parameter. We perform ablation experiments to choose values of ξ and arrive at $\xi = 1$ for XADQN and XATD3 and $\xi = 3$ for XASAC. We omit the detailed ablation study for brevity, but

[3]Their implementations come from RLlib, an open-source library for RL agents. We developed the XAER Python library, which can be easily integrated into RLlib and provides XA facilities for obtaining XADQN, XATD3 and XASAC.

full plots and auxiliary results can be found (together with source code) at `https://github.com/proroklab/xaer`.

To evaluate the performance of XAER compared to traditional Prioritized Experience Replay in tasks with complex and exception-heavy regulations, we trained agents in various environments, as described in Section 12.2. These environments differ in their rule density and complexity. We trained each agent for 40 million steps.

We report our scores by analysing the learning curve of mean episode rewards. We divide this curve into 20 segments, with each segment containing 5% of the total 40 million steps. To determine our reported scores, we identify the best segment (with the highest median) for each agent, aiming to compare agents at their peak performance. This approach helps us assess the effectiveness of experience replay methods when agents are performing at their best. Instead of relying on the overall best scores, which could be influenced by chance, we found that examining statistics over a span of 2 million steps (5% of 40 million) offered a more reliable comparison.

The medians and interquartile ranges (25-75%) for the selected segments are presented in Table 12.1. Please note that these calculations are based on the best segment identified for each agent, not on multiple repetitions of the same process.

Results in Table 12.1 show that across all tasks and methods, XAER versions only lose to the Prioritised Experience Replay baseline against DQN/Rainbow in Grid Drive Easy by 0.4%. For Grid Drive Medium and Hard, XADQN with `HOW+WHY` explanations exhibit significantly higher performance (57% and 81%, respectively). `WHY` and `HOW+WHY` exhibit similar performance in Graph Drive, being bested by `HOW` in Medium and Hard *Sparse* cases only. Although `HOW+WHY` explanations have consistently good results across environments, the version without the simplicity heuristic exhibited consistently inferior results. Neither baseline SAC nor TD3 managed to learn a policy in Graph Drive Hard Sparse (our hardest environment). XATD3 also failed to learn a policy in this environment, but XASAC achieved positive results.

Building upon these observations, we conducted several Mann-Whitney U-tests, similar to the experiments detailed in Chapter II. The improvements over the baseline shown in Table 12.1 all proved statistically significant with p-values well below the 0.05 threshold. This outcome is not totally unexpected, as different training strategies should inevitably lead to distinct policies.

Table 12.1: ***Results of the experiments on XAER.*** *Median cumulative rewards after 4.0×10^7 steps for experiments on Grid Drive, Graph Drive, and Graph Drive with sparse rewards (SR). Darker cells indicate better results in the environment. Bold is the best in a row. Interquartile ranges (25%-75%) in brackets.*

DQN/Rainbow	*Baseline*	*XADQN-HOW*	*XADQN-WHY*	*XADQN-HOW+WHY*	*XADQN-HOW+WHY sans simplicity*
Grid Easy	**17.13 (16.02-18.03)**	14.84 (12.88-15.88)	13.68 (11.73-15.29)	14.7 (13.08-15.91)	14.88 (13.33-16.04)
Grid Medium	7.99 (7.05-8.9)	7.59 (6.7-8.59)	8.06 (7.17-9.09)	**11.62 (10.48-12.66)**	9.21 (7.79-10.46)
Grid Hard	1.99 (1.74-2.24)	1.97 (1.72-2.24)	1.75 (1.51-2.03)	**3.14 (2.73-3.62)**	0.95 (0.8 - 1.14)

TD3	*Baseline*	*XATD3-HOW*	*XATD3-WHY*	*XATD3-HOW+WHY*	*XATD3-HOW+WHY sans simplicity*
Graph Easy	75.48 (68.09-80.85)	0.0 (-0.02-0.02)	88.75 (83.29-94.44)	**103.72 (98.64-107.03)**	84.23 (79.28-89.2)
Graph Medium	75.48 (68.09-80.85)	41.31 (33.24-47.49)	64.8 (59.44-69.47)	**78.34 (73.21-83.07)**	69.36 (61.01-77.58)
Graph Hard	-0.01 (-0.03-0.0)	-0.01 (-0.03-0.0)	**20.65 (18.9-22.4)**	14.54 (13.17-16.12)	10.31 (8.84-11.68)
Graph Easy (SR)	**2.65 (2.28-2.93)**	-0.04 (-0.06-(-0.02))	2.61 (2.43-2.75)	2.55 (2.42-2.66)	2.47 (2.34-2.62)
Graph Medium (SR)	0.34 (-1.0-0.97)	-0.04 (-0.05-(-0.03))	2.54 (2.3-2.79)	**2.75 (2.58-2.96)**	1.84 (1.47-2.0)
Graph Hard (SR)	-0.03 (-0.05-(-0.02))	-0.04 (-0.05-(-0.03))	-0.04 (-0.06-(-0.03))	-0.05 (-0.06-(-0.03))	-0.05 (-0.6-(-0.04))

SAC	*Baseline*	*XASAC-HOW*	*XASAC-WHY*	*XASAC-HOW+WHY*	*XASAC-HOW+WHY sans simplicity*
Graph Easy	65.9 (59.04-72.94)	79.46 (71.72-88.46)	138.81 (133.0-144.05)	**141.11 (136.45-145.87)**	116.39 (110.36-120.9)
Graph Medium	65.78 (58.43-71.92)	76.61 (69.64-83.69)	**112.16 (105.87-119.1)**	111.4 (106.72-116.11)	97.81 (92.91-103.1)
Graph Hard	26.85 (24.43-28.66)	22.92 (20.61-25.03)	32.14 (29.93-34.49)	**32.58 (30.41-34.69)**	17.85 (13.82-20.56)
Graph Easy (SR)	3.57 (3.19-4.01)	3.07 (2.82-3.21)	4.82 (4.64-4.98)	**4.83 (4.58-5.09)**	2.01 (1.8-2.19)
Graph Medium (SR)	2.61 (2.26-2.85)	**2.66 (2.26-2.98)**	2.64 (2.53-2.75)	2.47 (2.33-2.55)	2.31 (2.17-2.45)
Graph Hard (SR)[4]	1.15 (1.03-1.27)	**1.53 (1.37-1.63)**	1.11 (1.01-1.23)	-0.09 (-0.12-(-0.07))	0.81 (0.65-0.94)

Our results indicate a significant benefit achieved via explanation-aware experience replay, in support of Hypothesis 7. In one case (TD3 Hard), XAER *enabled* an agent to learn altogether where it would otherwise fail. XAER allowed agents to learn in Medium and Hard difficulty settings, obtaining significantly higher rewards whilst having the same hyper-parameters and number of learning steps.

The choice of explanation type also affected results: when superior, `HOW+WHY` explanations exhibited larger margins of improvement over other XAER methods. In other cases, when bested by `WHY` explanations, the former maintained very close results, thus achieving consistently satisfactory results in most cases. Also importantly, although `HOW` explanations exhibited lower performance than other XAER counterparts in most environments, it is worth noting that `HOW` explanations do not require an explainer and could, in theory, be used in any environment. The consistency of `HOW+WHY` results suggests that the act of explaining may involve answering more archetypal questions, not just causal ones, as also hypothesised in

Chapter 3 (see Hypothesis 1; cf. Section 3.2).

The frequency and magnitude of rewards are essential factors to consider in XAER clustering. When negative rewards are more frequent (with a similar magnitude to positive rewards), and there are more negative than positive clusters, oversampling may cause the agent to tackle situations with negative rewards more frequently, preventing it from maximising cumulative rewards. This effect can be particularly pronounced with very sparse rewards, such as the ones seen in the sparse version of Graph Drive.

Intuitively, this is akin to the notion that if there are few opportunities to explain, one must choose their explanations well. The notion of *explanation engineering* surfaces as a mechanism to orient the learning agent through means of selecting which experiences (and explanations) are more critical to the task at hand employing *abstractions*. Being explainable by design, explanation engineering can be an intuitive and *semantically-grounded* alternative to reward engineering, as the *meaning* of the rewards matters just as their magnitude. Examples include increasing the number of positive clusters or organising clusters hierarchically.

With regards to *relevance*, if the cumulative priority of the state transitions of a whole cluster is low, it may indicate that the agent has already learned to handle the task represented by the cluster, so it may not need it as an explanation (thus being less relevant). If the cumulative priority is high, it could indicate a further need for additional explanations. The cluster might represent either non-generic or generic tasks. If the agent needs explanations for a generic task, it should also need them for a non-generic task. In that case, the generic task is prioritised over the non-generic. The benefits of inter-cluster prioritisation (*simplicity*) are higher in environments with more complex rule sets and proportional to the complexity of the culture [162]. This suggests that uniformly selecting an explanation type to replay is less beneficial than selecting the simplest and most relevant explanation.

This work foments diverse avenues for further investigation. For one, further experiments could include the development of explainer functions to evaluate the performance of `WHY` explanations in popular benchmarks. Additionally, future work may observe the effect of XAER with on-policy algorithms, such as PPO [182]. Moreover, the illocutionary effect of explanations deriving from further archetypal questions could be explored in advanced explanation engineering for experience clustering.

CHAPTER 13

Extension of Explanation-Awareness to Decentralised Multi-Agent Reinforcement Learning

Extending XAER to centralised (execution) MARL is straightforward since having a centralised meta-agent can be seen as a particular instance of single-agent RL. However, using XAER with decentralised (execution) MARL algorithms is a different story. In fact, as discussed in Section 11.4, using (prioritised) experience replay with decentralised MARL agents may hinder learning.

MARL suffers from non-stationarity and experience replay exacerbates that problem. Therefore, extending Explanation-Aware Experience Replay to MARL is non-trivial as it would be a source of non-stationarity. Moreover, decentralised MARL suffers from high dimensionality due to the number of transitions per episode increasing with the number of agents, making experience replay even more challenging. Especially when the

number of observations per episode is too many (this is typical in decentralised MARL), relevant state transitions in a relatively small or too big buffer might not be replayed. Although techniques such as Global Distribution Matching (cf. Section 11.4) may address this issue in single-agent RL, they might increase the problem of non-stationarity even further in MARL. In other words, high dimensionality and non-stationarity tend to erode sample efficiency directly, rendering experience replay under strict memory constraints useless. We would need more on-policy experience for handling non-stationarity while exploiting a lot of old and (probably) off-policy experience would be necessary to cope with high dimensionality.

On the one hand, to address the issue of non-stationarity in MARL, some techniques [224] try modelling non-stationarity in the objective of new MARL algorithms, while others [75] address the problem with new experience replay schemes. Foerster et al. [75] propose a couple of Prioritised Experience Replay strategies specific for DQN that enable agents to distinguish old from new state transitions. In particular, it assigns "age fingerprints" or lower weights to old state transitions. Amongst these strategies, according to the empirical results of [75], the most effective is *multi-agent fingerprinting*, consisting in an age fingerprint added to the experience in the replay memory. However, this technique seems unsuitable for high-dimensional observation spaces and small experience buffers, considering that in those cases, the buffer would be saturated only with new state transitions, rendering multi-agent fingerprinting useless. Another technique is that of Nicholaus and Kang [144]. It relies on a strategy for sampling from the buffer only the experiences that are neither too similar nor too different from on-policy state transitions. This technique is incompatible with usual (replay) memory constraints, eventually requiring an experience buffer whose size is proportional to the observation space. Furthermore, it does not consider the need for a global distribution matching.

On the other hand, a solution to the problem of high dimensionality is XAER (cf. Chapter 12). Indeed, XAER is designed around the idea that some types of state transitions are so rare that neither standard Prioritised Experience Replay nor Global Distribution Matching can replay them. Relevant but (initially) "rare" state transitions could be dropped out of the buffer too soon without the agent learning from them. To address this issue and replay rare state transitions, XAER attaches an explanatory label e to transitions and, accordingly, partitions the experience buffer in prioritised clusters (one per different e) with constrained minimum and maximum load

from which experience is sampled in an unbiased way. Each one of these clusters represents a set of experiences explaining an uncommon situation beneficial to learning. So if one of these experience clusters is large enough, it can fully describe through a variety of examples, "`why` a specific reward is given", or "`how well` the agent is behaving compared to the average historical behaviour", etc. In other words, XAER is based on the intuition that oversampling those transitions capable of explaining uncommon situations is beneficial to learning.

In particular, the priority of a cluster is given by the sum of the priorities of its elements so that the oversampling ratio can be controlled by manipulating the maximum cluster size through ξ, where $\xi = 1$ means *full oversampling* and $\xi = \infty$ means *no oversampling*. In this sense, XAER has been proven to be beneficial with sparse rewards and relatively dense reward functions, improving the sample efficiency of several off-policy RL algorithms (i.e., SAC, TD3 and DQN) in a single-agent setting (cf. Chapter 12). For example, imagine that the experience buffer is like a book summarising the state space, and the agent has to read it. Suppose such a book is poorly organised and has too many pages (e.g., billions). In that case, it would be hard for the reader to finish it and memorise its content or find something useful by randomly selecting a few pages to read. On the contrary, if the book had only one page or thousands of pages containing the same information, the reader would not be able to learn much about the state space. In this sense, XAER is a mechanism to build a good summary of the state space. It is designed to help the agent acquire the most relevant minutia through explanation-aware experience oversampling.

However, XAER might strengthen the issues related to non-stationarity on decentralised MARL problems. Thus, to address this specific problem, we show how to combine XAER with Global Distribution Matching. Consequently, we propose DEER, an extension of XAER to MARL. This chapter will explain how DEER works and how it addresses the decentralisation-related issues mentioned above through a set of strategies to cope with the combinatorial explosion of problem complexity growing with the number of learning agents and the non-stationarity introduced by continuously evolving policies in decentralised settings. Just like XAER, also DEER is algorithm agnostic. This means that (differently from multi-agent fingerprinting [75]) it can work with any off-policy RL algorithm. Therefore, to prove that DEER is sufficiently generic, we tested it with DQN and SAC on typical MARL problems involving decentralisation. We release

the source code of DEER and the new environments used for the experiments under MIT license at `https://github.com/Francesco-Sovrano/DEER`.

13.1 DEER: Dimensionality-invariant Explanatory Experience Replay

Applying *experience replay* (especially the prioritised version) to MARL has several drawbacks. Primarily, it introduces non-stationarity by replaying old experiences that do not reflect the dynamics of the environment due to the agents' policies changing over time, thus violating the Markov assumption. Secondly, experience replay in order to be effective usually requires a buffer (of a fixed size) that is representative of the problem. Though, if the observation space is too large, the buffer might not be able to contain all the important experiences. This issue can be summarised by the following **exploitation problems**:

EP1. The experience buffer is relatively too small in proportion to the size of the observation space, thus losing correspondence to the real distribution of state transitions. Eventually, this might cause the agent to learn biased expected values by replaying only on-policy or meaningless experience;

EP2. The number of independent agents is too large, producing so many different state transitions in an episode to cover the entire capacity of the buffer or a significant fraction of it;

EP3. There are sparse rewards: these might be heavily under-sampled or even dropped out of the buffer without ever being replayed;

EP4. Reward engineering is employed: reward shaping introduces secondary rewards that can be replayed too frequently, distracting the agent from optimising the main objective.

So, let us suppose that we have a function that rewards n MARL agents for different reasons. What we could do to prevent reward feedback from being dropped from the buffer could be to organise it into clusters of experience. For example, one for each type of reward or reason, as in XAER, thus addressing the *exploitation problems* 3 and 4. In particular, if we consider XAER implementation, we can control how frequently certain rewards are

replayed through the hyper-parameter ξ. ξ defines the minimum and maximum size of clusters, where $\xi = 1$ means *full oversampling* (all clusters have the same size), $\xi = 2$ means *moderate oversampling* (there can be clusters whose size is twice the minimum size), and $\xi = \infty$ means *no oversampling* (the size of clusters has no constraints).

However, using experience over-sampling in combination with multi-agency might stress even more the non-stationarity issue. In addition, even if over-sampling could globally improve the correspondence of the buffer content to the real distribution of state transitions (by guaranteeing a certain diversity in the buffer of experience), it could not do much about the correspondence at a cluster level, leaving the *exploitation problems* 1 and 2 unaddressed. In particular, any solution to *exploitation problem* 2 must go through a mechanism of dropping experiences from the buffer that does not depend on insertion time. In this sense, a naive alternative strategy could be to consider *prioritised dropping* so that whenever the buffer is complete, the transition with the lowest priority is removed instead of the oldest one as in vanilla Prioritised Experience Replay. However, this approach may further emphasise the *exploitation problem* 1.

We designed DEER to guarantee a stationarity-aware and dimensionality-invariant correspondence of clusters to the real experience distribution. In particular, DEER applies to each cluster the Global Distribution Matching strategy of [98], but on a training-step basis. More specifically, DEER changes the way XAER drops experience from the buffer, it does so in a different way from *prioritised dropping*, thus mitigating both *exploitation problems* 1 and 2 together with problems 3 and 4.

As shown in Figure 13.1, when a new state transition is created and assigned to a cluster of a full buffer, instead of dropping the oldest state transition or the state transition with the lowest priority (from the clusters having reached the minimum size defined by ξ), DEER assigns to state transitions a random (but constant) drop probability d together with a *stationarity score* ρ, removing the state transitions having the lowest ρ and d (in this order). More specifically, the *stationarity score* ρ is the number of training steps σ (preceding the creation of the state transition) divided by the *stationarity window size* ϕ, as follows: $\rho = \lfloor \frac{\sigma}{\phi} \rfloor$.

We call this technique *stationarity-aware real distribution correspondence*. In fact, by choosing ϕ, it is possible to pragmatically control the amount of non-stationarity within the experience buffer whilst enforcing a proper correspondence to the real distribution of state transitions thanks to

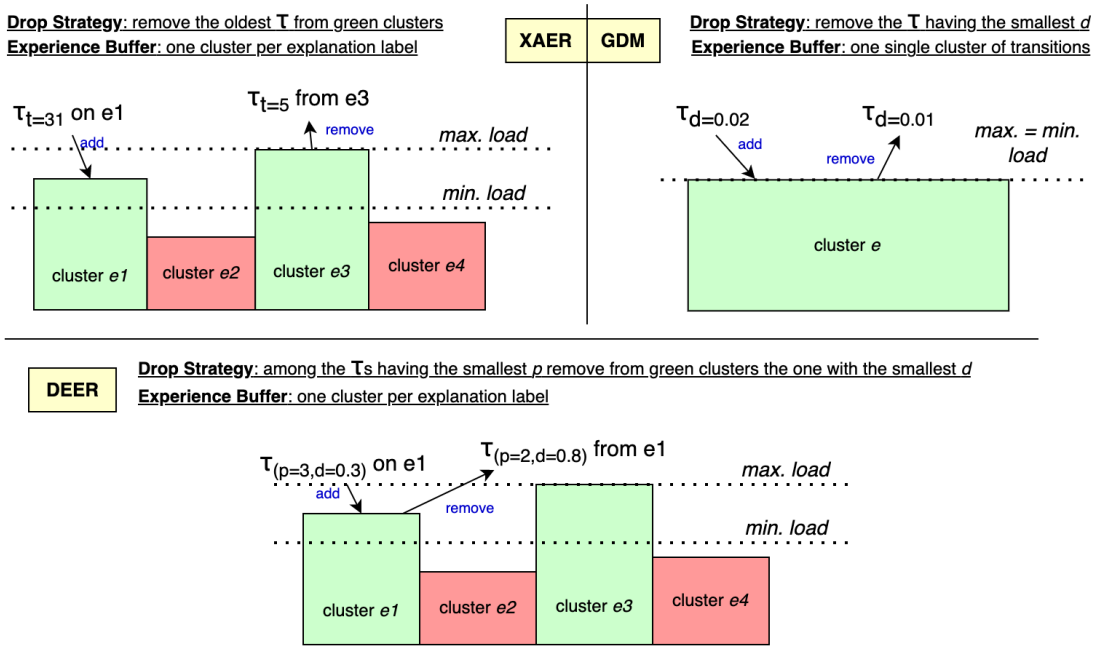


Figure 13.1: ***Main differences between Global Distribution Matching, XAER and DEER.*** *This diagram shows how experience is dropped from clusters (the coloured rectangles). Red clusters are those from which no experience can be dropped because they do not contain a sufficient number of state transitions (i.e., the minimum load), regardless of the drop priority or insertion time. In DEER, when a state transition $\tau_{(\rho,d)}$ is added to a full cluster (i.e., a cluster with maximum load), the state transition having the lowest d amongst those with the lowest stationarity score ρ is dropped from green clusters. d is a random number assigned once to the state transition. In XAER, the oldest state transition τ_t of the green clusters is dropped instead. t is the insertion time of τ. On the contrary, in Global Distribution Matching (GDM for short), the state transition having the lowest d is dropped.*

the random assignment of d, as explained in [98]. In this sense, DEER improves over XAER and Global Distribution Matching, ensuring the correct functioning of experience replay when scaling the number of agents and the size of the observation space.

In particular, setting $\phi = \infty$ is equivalent to vanilla distribution match-

ing (which does not take non-stationarity into account), while $\phi = 1$ means that stationarity heavily changes at every training step. Instead, a $1 < \phi < \infty$ (e.g., $\phi = 5$) means that stationarity changes smoothly so that, every ϕ training steps, the content of the experience buffer has to be replaced with new state transitions. Therefore, the more frequently the stationarity is deemed to change during training, the lower the ϕ should be.

13.2 Environment for Evaluating DEER

Generally speaking, decentralised MARL problems can happen in discrete (e.g., grids) or continuous environments (e.g., graphs), on simple or complex scenarios (e.g., a real city governed by strict regulations), with discrete (e.g., go to the grid cell on the left) or continuous actions (e.g., turn 30° to the left), with homogeneous or heterogeneous agents, with holonomic or non-holonomic constraints.

We focus on problems of decentralised multi-agent pathfinding and decentralised task assignment under uncertainty. On the one hand, the problem of *multi-robot task assignment under uncertainty* is strongly NP-hard [158] and an important research topic in multi-agent systems [136]. It is defined as allocating tasks to agents that minimise an uncertain allocation cost (e.g., time). So far, no generic poly-time algorithm is known to solve it efficiently [158]. An example of *task assignment* problem is that of a swarm of bots that have to coordinate for delivering items to multiple targets on a planar graph (e.g., a city). On the other hand, also *multi-agent pathfinding* is an NP-hard problem, even when approximating optimal solutions [177]. In particular, multi-agent pathfinding is an instance of multi-agent planning. It calculates collision-free paths for a group of agents from their position to an assigned target. An example of *pathfinding* problem is the one used by Sartoretti et al. to evaluate PRIMAL [177].

What is common to these problems is that they all are NP-hard in the most generic case due to the complexity of dealing with partial observability caused by decentralisation and uncertainty in the environment. In other words, as anticipated in Section 11.4, such problems typically require addressing partial observability, non-stationarity and high dimensionality issues.

To better understand the nature of the high dimensionality problem, we hereby present a formal analysis of the typical dimensionality of the observation space in a graph problem of decentralised task assignment under

uncertainty. In particular, let us assume that the problem is instantiated on a planar graph of V nodes and E edges having different lengths. In this scenario, a task is for an agent a to navigate through a sub-set of edges for delivering an item (e.g., a medicine) to a target node $t \in T$ (e.g., a person). Moreover, let us assume that all agents have full visibility of the graph but not of the other agents, and let us define:

- A as the number of agents;
- $\bar{V} \leq V$ as the average number of nodes that an agent can reach;
- $\bar{E} \leq E$ as the average number of edges that an agent can traverse;
- λ as the average length of the $\bar{E}$ edges.

We have that the number of possible observations depends on the number $C_v = \sum_{k=1}^{A} \binom{\bar{V}}{k}$ of possible combinations of agents on nodes and the number $C_e = \sum_{k=1}^{A} \binom{\lceil \lambda \bar{E} \rceil}{k}$ of possible combinations of agents on edges. An agent can be observed either on a node or edge. Thus, considering that the number of points of all visible edges is approximately $\lceil \lambda \bar{E} \rceil$ and that the number of visible nodes is $\bar{V}$, it follows that the size of the observation space is directly proportional to $C_v + C_e$, as a partial sum of binomial coefficients that cannot be captured in a closed form and that grows combinatorially in the number of agents.

To evaluate the effects of DEER in a diverse configuration space of environments with different dimensionality issues, we developed modular environments that allow us to systematically change its properties. These **environments** are namely:

- **Grid Planning**: a grid-like environment compatible with DQN simulating multi-agent planning problems, where agents can take discrete actions (e.g., move left, right, up, down).
- **Graph Delivery**: a graph-like environment compatible with SAC simulating multi-robot task assignment problems, where agents can take continuous actions (e.g., steer by 30°).

13.2.1 Graph Delivery: Decentralised Task Assignment on Graphs

To test and validate DEER on problems of *decentralised task assignment under uncertainty*, we created Graph Delivery. Graph Delivery is a new

configurable Gym [31] environment that produces instances of multi-robot task assignment under uncertainty (a visualisation is shown in Figure 13.2) on randomly generated planar graphs. More specifically, the planar graphs are built in poly-time[1] from randomly generated Euclidean minimum spanning trees [82] extended with Delaunay triangulations [57], given as input the desired number of nodes, the maximum number of edges per node, the minimum distance between two nodes and the maximum width and height of the graph.

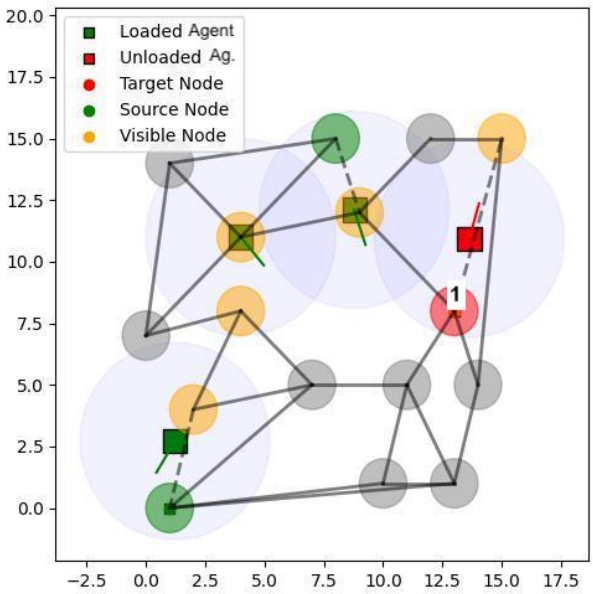


Figure 13.2: ***Screenshot of Graph Delivery.*** *Visualisation of a randomly generated instance of Graph Delivery with* 4 *agents under complete partial observability, spawned on* 2 *sources of a graph of* 16 *nodes (having a minimum distance of* 2.5 *from each other) and* 1 *target with capacity* 2*, in a* $16x16$ *map with maximum* 4 *edges per node. Agents are represented with coloured squares, their orientation with a heading vector connected to the square, while the payload of targets is given above the node. The field of view of agents is shown with a circular blue shade.*

In Graph Delivery, A agents have to coordinate for delivering items (e.g., food, medicines) from a set S of *source nodes* to a set T of *target nodes*, in the minimum possible amount of time. Moreover, agents can be

[1] $O(n \log n)$, where n is the number of nodes.

loaded (with an item) or *unloaded*, and they can be spawned in any node of the graph. Agents are initially load free; whenever they reach (or are spawned on) a source node, they become *loaded*. Each agent can carry only one item at a time and bring it to one target $t \in T$ until the target receives a pre-defined number N_t of deliveries. Hence, the goal in Graph Delivery is for agents to bring the N_t required deliveries to all $t \in T$ as quickly as possible.

Actions. The action space of Graph Delivery is continuous, with agents that can only steer (with infinite angular velocity) when on nodes or decide whether to go forward or backward (when on edges). Thus, DEER can be tested with Graph Delivery only on RL algorithms which admit continuous action spaces such as SAC or TD3. In particular, the steering angle chosen by an agent can be any real number in $[0, 360]$. Agents can move through other agents and might get stuck in a node if they do not steer correctly.

Heterogeneity. Graph Delivery allows casting the planar graphs into road networks (as Graph Drive; cf. Section 12.2.2). It attaches regulations to them and properties to agents and edges (i.e., the roads). Heterogeneity, in this case, is measured in terms of the agent's properties that impact the agent's functionality and ability to play a role in deliveries. In other terms, heterogeneity is given by: *i)* roads with different characteristics and requirements; *ii)* agents with different features. In particular, we implemented two **levels of heterogeneity**.

Null Heterogeneity. All agents are equal, and all roads are the same; no heterogeneity.

Simple Heterogeneity. Agents and roads may possess a combination of 3 properties each. The road network is governed by four rules indicating which properties of agents are compatible with which properties of roads. In particular, agents can be emergency vehicles, have special permission, and be able to pay a fee. Instead, roads can have an accident, require special permission, and require to pay a fee. So, the four rules are: *i)* only emergency vehicles can pass through roads with accidents; *ii)* only vehicles with special permissions can drive along routes requiring special permissions; *iii)* only if an agent can pay fees, or it can pass through an edge requiring to pay a fee; *iv)* if an agent is an emergency vehicle, it does not have to pay any fee.

Rewards. Rewards are given at every step to each agent separately. We designed two reward functions for Graph Delivery to test how DEER be-

haves with different reward frequencies. The first reward function is sparser and as follows:

$$\begin{cases} +1 \text{ (terminal)} & \text{when the agent delivers the last delivery} \\ +1 & \text{when the agent unloads} \\ 0 & \text{otherwise} \end{cases}$$

The second function gives rewards more frequently, as follows:

$$\begin{cases} +1 \text{ (terminal)} & \text{when the agent delivers the last delivery} \\ +1 & \text{when the agent unloads} \\ +1 & \text{when the agent get loaded} \\ -1 & \text{when the agent violates the regulation (if any)} \\ -1 & \text{when the agent gets stuck in a node} \\ 0 & \text{otherwise} \end{cases}$$

Observations. The size of the observation space depends on the level of heterogeneity. Agents can observe their state and the whole graph as an ordered sequence of information about nodes. The agent's state consists of position, loading status, global task completion rate, and heterogeneity features (if any). The information about nodes consists of coordinates, the number of received deliveries (if the node is a target), and attached edges. Information about edges consists of the coordinates of their ends and heterogeneity features (if any). Agents can only see and communicate with other agents within a fixed radius, receiving fixed-size messages (e.g., 72 bytes). In other words, agents can only sometimes get access to the position or state of other agents. This partial observability introduces *uncertainty in task allocation times* whenever agents are initially spawned on different (source) nodes. That is because, in that case planning an optimal assignment would be impossible with the information agents have at their disposal at step 0. Even if travel times are deterministic, in this situation, agents could not know if others will finish a target $t \in T$ before them. This is the source of task assignment uncertainty.

Explanations. `WHY` explanations are attached to state transitions in a similar way to Graph Drive (cf. Section 12.2.2). However, there are some differences. The following explanations are attached to an agent's state

transition:

$$\begin{cases} \textit{“has just delivered”} & \text{when the agent is unloaded} \\ \textit{“has just taken”} & \text{when the agent is loaded} \\ \text{the violated rules} & \text{when the regulation is violated} \\ \textit{“stuck”} & \text{when the agent is stuck in a node} \\ \textit{“is on node”} & \text{when the agent is on a node} \\ \textit{“is moving”} & \text{otherwise} \end{cases}$$

To summarise, what makes graph delivery suitable for testing MARL algorithms on the problem of “decentralised task assignment under uncertainty” is the fact that it supports:

- **Heterogeneity**: graphs and deliveries can be constrained by regulations of different complexity.
- **Uncertainty and decentralisation**: there can be uncertainty in the position of other agents.
- **Infinite problem instances**: with Graph Delivery, it is possible to simulate, in a scalable way, realistic problems of task allocation with continuous actions, on planar graphs of any size and density, with edges of variable length and any arbitrary number of agents.

13.2.2 Grid Planning: Multi-agent Pathfinding on Grids

Grid Planning is a discrete environment for testing DEER with DQN on multi-agent pathfinding problems. Specifically, Grid Planning is a 2D discrete 4-connected grid world, the same environment used by Sartoretti et al. to test their PRIMAL2 algorithm [54]. In Grid Planning, a set A of agents has to coordinate for optimally planning the shortest path to reach a predefined goal position in a grid of size 20×20. The grid is a maze that contains walls (i.e., of a maximum length of 20). Walls are obstacles, i.e., non-traversable grid cells. The density of obstacles can be controlled via a hyper-parameter. We set it to 0.3. When an agent reaches its goal, it terminates the episode, while the other agents can continue to pursue their goals. In other words, the version of Grid Planning we considered does allow for lifelong learning. For examples of Grid Planning and more details about this environment, read [177, 54].

Actions. The action space of Grid Planning is discrete as Grid Drive (cf. Section 12.2.1). Agents can only move to a neighbouring location left, right, top, down, or stay still. At each time step, agents can only perform one action and cannot move through walls or other agents.

Rewards. The reward function we adopted is different from [54]. Rewards are given at every step to each agent separately, following these criteria:

$$\begin{cases} +1 \text{ (terminal)} & \text{when the agent reaches its goal} \\ 0 & \text{otherwise} \end{cases}$$

Observations. Grid Planning is a partially observable grid world. Agents can see the location of their goal, wherever they are. Moreover, agents can also observe the state of the world in a limited field of view (in practice, 11×11) centred around themselves. In this limited field of view, information is separated into eleven channels to aid learning:

> *"Four binary [channels] provide information about obstacles, positions of other agents, goals of those observable agents, and the agent's own current goal position if within the [field of view]; three scalar values provide each agent with a unit vector pointing towards its goal and the absolute magnitude of the distance to its goal at all times. [...] A path length [channel provides information about] the (normalised) shortest-path distance to [the] goal from each non-obstacle cell. These distances are calculated using single-agent A*[2], ignoring all other agents in the environment. [...] Three smaller spatial [channels provide] information about neighbouring corridors. [...] [Three more channels contain the predicted future position of other agents within its local field of view. Each one of these last channels] refers to the number of future time steps that an agent looks ahead to. For each time step, the predicted future position of all visible neighbouring agents at that time step is shown on the map. These maps are generated using single-agent A*."[54]*

Explanations. `WHY` explanations are attached to state transitions in a similar way to Grid Drive (cf. Section 12.2.1). We use the A* pathfinding

[2] A* is a single-agent graph traversal, and pathfinding algorithm [173].

algorithm [173] to generate them. `WHY` explanations are:

$$\begin{cases} \textit{“invalid action”} & \text{if the agent's action is invalid} \\ \textit{“acting as A*”} & \text{if the agent is acting as A* would} \\ \textit{“no path”} & \text{if there is no path left for the agent} \\ \textit{“acting differently”} & \text{if the agent is acting differently from A*} \end{cases}$$

13.3 Evaluation of DEER and Results Discussion

DEER is designed to extend XAER to decentralised MARL. To show that DEER is better than the baselines, we devised a few experiments on different configurations of Graph Delivery and Grid Planning. We want to show that DEER can cope with non-stationarity better than the baselines. This is done by comparing the performance of DEER when changing the number of agents and the reward function in Grid Planning, Graph Delivery and Graph Delivery with *Simple Heterogeneity*. Considering that multiple reward functions are involved in some of the experiments, performance scores in Graph Delivery are measured in terms of *completed deliveries* (the higher, the better) instead of cumulative rewards. In contrast, in Grid Planning, performance scores are measured in cumulative rewards.

On the one hand, Graph Delivery experiments were run with SAC using the following configuration: a 40×40 map; maximum 4 edges per node; 40 nodes having a minimum distance of 2.5; $|A| = \{15, 21\}$ independent agents; $|T| = \lfloor \frac{|A|}{3} \rfloor$ targets with capacity $\forall t \in T : N_t = 2$; and $|S| = 2$ different sources; agents spawned only on sources; a visibility radius of 8; and $\tau = 3e^{-5}$ (τ is a hyper-parameter specific to SAC).

On the other hand, Grid Planning experiments were run with DQN/Rainbow using the following environment configuration: a 20×20 grid; $|A| = \{16, 20\}$ independent agents; a maximum wall length of 20; a field of view of 11; and an obstacle density of 0.3.

Importantly, we used a centralised experience buffer with a capacity of 2^{12} state transitions and episodes of maximum 2^8 steps. We did it to exaggerate the exploitation problems 1 and 2 (cf. Section 13.1). In fact, the minimum number of agents in Graph Delivery and Grid Planning is about 2^4. So, the state transitions of one episode (which are *steps per episode* × *number of agents*) are enough to fill the entire buffer.

Other important hyper-parameters common to all experiments were: training batches of 2^8 state transitions, $\gamma = 0.999$, $\xi = 2$ and “*complete*

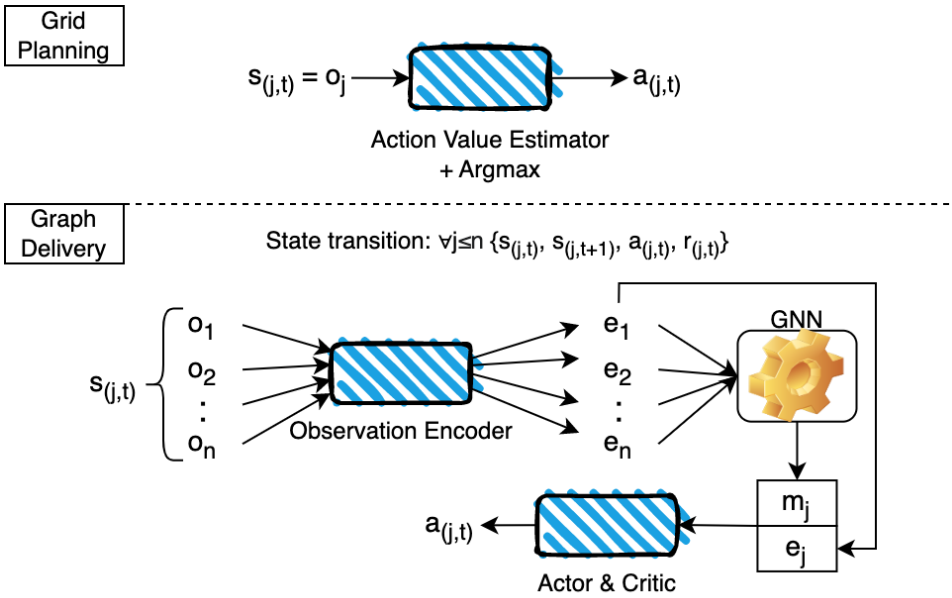


Figure 13.3: ***Sketch of the neural network architectures used for Graph Delivery and Grid Planning.*** *On the top, we see that a vanilla architecture is used for Grid Planning. For each agent j at step t, this architecture consists of a fully connected layer which takes as input the state $s(j,t)$ (which consists only of what j observes). Then it outputs the best-estimated action $a(j,t)$ together with its estimated value (which we do not show here). On the bottom is the (more complex) architecture used for Graph Delivery. Here, state $s(j,t)$ contains the observations of all agents visible to j. These are encoded by a fully connected layer and aggregated by a GNN to create the message m_j for j. Then, message m_j is concatenated with the observation of j embedded by the observation encoder. The resulting vector is then given to the actor and critic to produce $a(j,t)$ (together with the state-value, which we do not show here). Both the actor and the critic are fully connected layers.*

episodes" as *batch mode* (this means that transitions are not inserted in the experience buffer until the episode ends).

Additionally, Graph Delivery has agents that communicate in a decentralised fashion with messages of fixed size. Thus we implemented the communication channel as a differentiable GNN (as it is typically done in these cases; cf. Section 11.4). In particular, the neural network architec-

Table 13.1: ***Results of experiments on XAER and DEER in multi-agent problems.*** *This table offers a comparison of the median cumulative rewards after 100 million steps from experiments conducted on three scenarios: Grid Planning, Graph Delivery, and Graph Delivery with heterogeneity (termed as Hetero). The methods compared are DEER, DEER with $\phi = 10$, XAER, and Prioritised Experience Replay (abbreviated as PER). The Grid Planning experiments were conducted with either 16 or 20 agents, while the Graph Delivery experiments involved either 15 or 21 agents. Additionally, two distinct reward functions were used in the Graph Delivery scenario: sparse and frequent (as detailed in Section 13.2.1). In this table, darker cells highlight superior results for a given environment. The best result in each row is highlighted in bold. Furthermore, the table includes interquartile ranges (spanning from the 25th percentile to the 75th percentile) enclosed within brackets.*

Grid Planning	PER	XAER	DEER ($\phi = 5$)	DEER ($\phi = 10$)
16 Agents	1.55 (1.45 - 1.65)	4.66 (4.34 - 4.88)	4.83 (4.57 - 4.99)	**5.09 (4.92 - 5.28)**
20 Agents	1.96 (1.87 - 2.05)	6.89 (6.53 - 7.17)	**7.04 (6.71 - 7.32)**	5.82 (5.47 - 6.23)

Graph Delivery	PER	XAER	DEER ($\phi = 5$)	DEER ($\phi = 10$)
15 Agents / Sparse	5.01 (4.9 - 5.13)	8.28 (8.13 - 8.4)	**8.35 (8.25 - 8.44)**	8.25 (8.13 - 8.32)
21 Agents / Sparse	7.84 (7.67 - 7.98)	11.84 (11.73 - 11.98)	11.94 (11.77 - 12.2)	**12.0 (11.86 - 12.2)**
15 Agents / Freq.	4.97 (4.82 - 5.09)	8.64 (8.49 - 8.75)	8.65 (8.56 - 8.72)	**8.71 (8.63 - 8.8)**
21 Agents / Freq.	7.73 (7.54 - 7.91)	12.55 (12.43 - 12.64)	**12.65 (12.52 - 12.76)**	12.6 (12.47 - 12.7)

Graph Delivery (Hetero)	PER	XAER	DEER ($\phi = 5$)	DEER ($\phi = 10$)
15 Agents / Sparse	3.33 (3.22 - 3.42)	4.81 (4.66 - 4.89)	**5.27 (5.04 - 5.43)**	5.02 (4.84 - 5.14)
21 Agents / Sparse	5.03 (4.87 - 5.16)	7.34 (7.2 - 7.49)	**7.55 (7.24 - 7.77)**	7.44 (7.22 - 7.7)
15 Agents / Freq.	5.18 (5.07 - 5.29)	5.37 (5.28 - 5.45)	**5.39 (5.26 - 5.51)**	5.36 (5.23 - 5.47)
21 Agents / Freq.	5.92 (5.67 - 6.25)	8.03 (7.89 - 8.17)	8.01 (7.87 - 8.17)	**8.06 (7.88 - 8.17)**

ture used for Graph Delivery is shown (and explained) in Figure 13.3. As GNN we used a Graph Attention Network called GATv2Conv[3] [32]. Conversely, the neural network architecture used for Grid Planning (also shown in Figure 13.3) did not involve any GNN or communication channel.

To simplify training without sacrificing decentralisation at runtime, we adopted the *centralised training and decentralised execution* strategy. We did it through *independent learning*, where the same policy is used for all the agents through parameter sharing without requiring a (hard-to-train)

[3] `https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html?highlight=GATv2Conv`

Table 13.2: ***Improvement Rates.*** *This table shows how much in percentage DEER improves over XAER and Prioritised Experience Replay (PER). The "Fewer Agents" column means 16 agents for Grid Planning and 15 for Graph Delivery. Instead, the "More Agents" column means 20 agents for Grid Planning and 21 for Graph Delivery.*

	DEER over PER		DEER over XAER	
	Fewer Agents	More Agents	Fewer Agents	More Agents
Grid Planning	+228.3%	+259.1%	+9.2%	+2.1%
Graph Delivery Sparse	+66.6%	+53%	+0.8%	+1.3%
Graph Delivery Freq.	+75.2%	+63.6%	+0.8%	+0.7%
Hetero Graph Delivery Sparse	+58.2%	+50%	+9.5%	+1.6%
Hetero Graph Delivery Freq.	+4%	+36.1%	+0.3%	+0.3%

joint action space. For partitioning the experience buffer with XAER and DEER, we used both `WHY` and `HOW` explanations (cf. Section 12.1.1). For DEER we used two *stationarity window sizes* $\phi = 5, 10$, thus studying how ϕ impacts on performance.

Training in all experiments was performed for 10^8 environment steps, with a random seed of 42. As with XAER (cf. Section 12.3), our reported scores are obtained by segmenting the curve of mean episode deliveries into 20 regions containing 5% of steps each. We select the best region (highest median) for each agent to compare agents at their respective best performances. We report those medians in Table 13.1, as well as the 25-75% inter-quartile range for the selected region.

The empirical results once again lend support to the argument that both XAER and DEER enhance the sample efficiency of off-policy RL algorithms. This is in alignment with Hypothesis 7. XAER and DEER indeed outperform the baseline, showing average improvements of over 200% for Grid Planning and over 50% for Graph Delivery. Furthermore, as anticipated, DEER outperforms XAER, although the performance gap between the two is relatively small; around a 5% improvement for Grid Planning and about a 1% improvement for Graph Delivery. These findings suggest that the `HOW` and `WHY` clustering strategies deployed by XAER might already

encapsulate, to some extent, the global distribution of state transitions. In a similar vein to our previous method in Section 12.3, we conducted several Mann-Whitney U-tests. Consistently, the improvements over prioritized experience replay, as illustrated in Table 13.1, all proved to be statistically significant with p-values well below the 0.05 threshold.

The non-stationarity mitigation technique employed by DEER is straightforward yet effective. Nevertheless, managing non-stationarity optimally is not trivial; we have just scratched the tip of the iceberg. For example, the approach followed by DEER does not consider the magnitude of policy changes at each training step, relying solely on *constant stationarity scores* that do not adjust for training events. In this sense, implementing a system for generating adaptive *stationarity scores* could be the next step forward, but we leave it as future work. Alternative and improved solutions compatible with DEER and XAER could, for example, be based on non-stationarity-aware clustering strategies to construct different experience buffers for each cooperation/defection strategy under the assumption that each strategy retains some degree of stationarity. In other words, more sophisticated strategies may be needed to deal optimally with non-stationarity. We believe that, in this regard, DEER has several tools and a high degree of flexibility.

Conclusion

THE MAIN objective of this dissertation was to produce new theories, models, algorithms and tools for generating user-centred and goal-driven explanations from large and heterogeneous collections of explainable information. We achieved the objective in several ways.

To demonstrate that the identified theories are generic enough to broadly capture the nature of explanations, we tested them not only with humans but also with AI agents. Indeed, we have conducted various user studies (involving hundreds of human subjects) and experiments, showing that our proposed user-centred explanatory process model is generic enough to benefit both human and artificial intelligence. Our technology has been able to produce more effective and satisfying explanations for various explananda (including educational textbooks, software documentation and complex regulations), improving the state of the art of Reinforcement Learning and Human-Computer Interaction.

Specifically, this dissertation was built around the following research questions:

RQ1. How can one define *explaining*, *explanations* and *explainability*?

RQ2. How to quantitatively evaluate *explanations* and *explainability*?

RQ3. How to model an automatic (user-centred) *explanatory process*?

RQ4. How to algorithmically generate *explanations* for humans?

RQ5. Would a better understanding of what constitutes an *explanatory process* help improve artificial intelligence (i.e., machine learning)?

In order to justify and evaluate the proposed theories and models, we considered case studies and examples in the intersection of AI and law, focusing in particular on European legislation. We started our journey from the existing requirements and common concepts of explanations provided by the law, focusing mainly on the work produced by the European Commission and its expert groups. Analysing the GDPR and the ethical guidelines of the High-Level Expert Group on Artificial Intelligence, we found that user-centred explanatory tools are considered an essential ingredient for reliable AI (cf. Chapter 1). This finding prompted us to focus on user-centred explanations and to consider *explainable information* and *explanations* as two different things. We conducted an exploratory review of contemporary theories of explanation in philosophy, looking for those compatible with a user-centred view of explanations that could be practically implemented in a software application. In this sense, our work was inspired by Miller [138], because it tries to reconcile AI, human-computer interaction, philosophy and law.

We identified Achinstein's theory from Ordinary Language Philosophy as a suitable (and understudied) candidate, which frames the act of explaining as an illocutionary (i.e., broad but relevant and deliberate) act of pragmatic question answering (cf. Chapter 2). Then we expanded and adapted Achinstein's theory to our needs, identifying usability metrics as a suitable way to evaluate explanations (cf. Chapter 3). Consequently, we also formally defined *explanatory illocution* as the primary mechanism responsible for the anticipation of unasked (archetypal) questions, proposing a mathematical formula to quantify the Degree of Explainability (DoX) of textual information on top of that (cf. Chapter 4), thus answering *RQ1* and *RQ2*. Importantly, DoX is the first metric based on Ordinary Language Philosophy to quantify explainability objectively.

In order to answer *RQ3*, we delved into the differences between explainable information and explanations. We suggested considering as separate things *how information is made explainable* (e.g., through XAI algorithms) from *how explainable information is selected and organised into explanations* (e.g., through Explanatory Artificial Intelligence algorithms). We then gave a formal definition of the explanatory process as a function capable of decomposing the space of all possible explanations (or explanatory space) in a tree-like structure for efficient exploration by an explainee. We

also explained how existing linguistic theories could be used to represent explanatory spaces as hypergraphs of questions and answers. Next, we proposed a set of heuristics (the ARS heuristics) for decomposing an explanatory space in a user-centred manner as well as a set of commands (the SAGE commands) to explore it through question-answering (cf. Chapter 5). Specifically, these heuristics are designed to help users follow their drifts of interest as they explore the explanatory space.

We then answered *RQ4* by creating a YAI for humans (YAI4Hu, for short; cf. Chapter 6), an implementation of the SAGE-ARS model based on AI for question-answering. YAI4Hu was tested with several user studies involving hundreds of human subjects from different user groups. By comparing YAI4Hu with several baseline explanatory tools, we showed that not all explanatory space decompositions are equally helpful for humans and that our SAGE-ARS model can produce more usable explanations (cf. Hypothesis 3; Section 5.4). We also provided empirical evidence that explanatory illocution involves answering archetypal questions (cf. Hypothesis 1; Section 3.2).

Eventually, we showed how to use our technology to design explanatory software compliant with the European GDPR (cf. Section 5.5). We also discussed how to assess the compliance of software documentation with Business-to-Consumer and Business-to-Business requirements established by European legal provisions (cf. Section 8.4) or how to identify explainability metrics which could ease the assessment of compliance with the proposed European AI Act (cf. Section 4.2).

We also further tested Hypothesis 1 (i.e., "explanatory illocution is about answering archetypal questions"), providing empirical evidence that intelligently anticipating implicit questions helps produce better explanations for humans. To this end, we tested our YAI on educational tasks with a user study involving more than one hundred English-speaking students. We assumed that the writer of an educational text tries, for narrative purposes, to explain best the most important topics at hand, thus (according to our theory) implicitly identifying the essential questions whose answers provide a good overview of the topics. Based on this assumption (cf. Hypothesis 6), we created YAI for education (YAI4Edu, for short; cf. Chapter 10), an intelligent interface that extends YAI4Hu to improve the explanatory power of the excerpts of an educational textbook for teaching how to write a legal memorandum. In particular, YAI4Edu uses DoX and a couple of new strategies (SyntagmTuner and DiscoLQA; cf. Chapter 9) to spe-

cialise question-answering on legal English, together with an algorithm for the automatic extraction of archetypal questions in order to produce more intelligent explanations.

As an answer to *RQ5*, we showed that the SAGE-ARS model could produce more user-centred explanations not only for humans but also for RL agents (cf. Hypothesis 7). In Chapter 12, we presented XAER, an RL algorithm that implements the ARS heuristics and can be used to explain complex road regulations and to improve the sample efficiency of seminal single-agent off-policy algorithms such as SAC, TD3 and DQN. In Chapter 13, we also showed how to extend XAER to a Multi-Agent Reinforcement Learning context by presenting DEER.

We demonstrated how explanations to humans and machines could be reduced to the same process of organising knowledge in clusters of answers to implicit archetypal questions, showing how to identify such questions. With YAI4Hu, we tested that organising explanations in terms of clusters of answers to generic archetypal questions (e.g., `what`, `how`, `why`, `who`) helps to produce better explanations. Instead, with YAI4Edu, we showed how to automatically extract less generic archetypal questions from the data by exploiting linguistic theories. Similarly, with XAER and DEER, we showed that organising the experience buffer of an RL agent in terms of clusters of state transitions that answer different (archetypal) questions (not only about causality) can drastically improve the agent's ability to absorb knowledge and learn a better policy sooner. With XAER, we also proposed the concept of explanation engineering as an alternative way to reward engineering for improving the performance of RL agents.

Explanations play an essential role in human society and are one of the fundamental mechanisms underlying our education and technological advances. There still needs to be more agreement on the definition of explanation and explainability, despite the joint efforts of many philosophers and scientists. However, our work has the potential to change this situation by creating a bridge between artificial intelligence and philosophy that will help improve both technology and theory.

Overall, we stress that the research of explainable and explanatory AI should emphasise a proper understanding of what constitutes the act of explaining. For this reason, we have reworked several ideas from Achinstein's theory of explanations. The critical link with usability is that explanations require illocution, i.e., answering the user's implicit questions. This re-

quirement needs to be considered by works that treat explanations only as a product, independent of the user's goals or knowledge.

Therefore, we can firmly conclude that whatever approach is used to describe an explanatory space, it should ensure that such a description is more expressive than bare XAI outputs and at least as expressive as an nth-level explanatory closure. Indeed, it should allow users to efficiently identify and create their own goal-driven narratives as (possibly) short paths within the explanatory space. Consequently, explaining is hard regardless of whether the receiver of explanations is a human or a machine. This is because defining the ARS heuristics for optimally exploring an explanatory space is generally not straightforward. Indeed, explaining to humans can be arduous, as it is only sometimes possible to correctly identify what is most relevant to a person. Similarly, explaining to RL agents can also be complicated, particularly by the complexity of framing useful abstractions for an agent. These challenges concretely shape the complexity of identifying user-centred decompositions of explanatory spaces.

Bibliography

[1] Agnar Aamodt and Enric Plaza. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7:39–59, 1994. doi: 10.3233/AIC-1994-7104.

[2] Peter Achinstein. *The Nature of Explanation*. Oxford University Press, 1983. ISBN 9780195037432. URL `https://books.google.it/books?id=0XI8DwAAQBAJ`.

[3] Peter Achinstein. *Evidence, explanation, and realism: Essays in philosophy of science*. Oxford University Press, 2010. ISBN 978-0199735259.

[4] B. Albert and T. Tullis. *Measuring the User Experience: Collecting, Analyzing, and Presenting Usability Metrics*. Interactive Technologies. Elsevier Science, 2013. ISBN 9780124157927. URL `https://books.google.it/books?id=bPhLeMBLEkAC`.

[5] Roohallah Alizadehsani, M Roshanzamir, Moloud Abdar, Adham Beykikhoshk, Abbas Khosravi, M Panahiazar, Afsaneh Koohestani, F Khozeimeh, Saeid Nahavandi, and N Sarrafzadegan. A database for using machine learning and data mining techniques for coronary artery disease diagnosis. *Scientific data*, 6(1):1–13, 2019. doi: 10.1038/s41597-019-0206-3.

[6] Dean Allemang and James A. Hendler. *Semantic Web for the Working Ontologist - Effective Modeling in RDFS and OWL, Second Edition*. Morgan

Kaufmann, 2011. ISBN 978-0-12-385965-5. URL http://www.elsevierdirect.com/product.jsp?isbn=9780123859655.

[7] Richard A Armstrong. When to use the bonferroni correction. *Ophthalmic and Physiological Optics*, 34(5):502–508, 2014.

[8] Leila Arras, Ahmed Osman, and Wojciech Samek. CLEVR-XAI: A benchmark dataset for the ground truth evaluation of neural network explanations. *Inf. Fusion*, 81:14–40, 2022. doi: 10.1016/j.inffus.2021.11.008. URL https://doi.org/10.1016/j.inffus.2021.11.008.

[9] Alejandro Barredo Arrieta, Natalia Díaz Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. *Inf. Fusion*, 58:82–115, 2020. doi: 10.1016/j.inffus.2019.12.012. URL https://doi.org/10.1016/j.inffus.2019.12.012.

[10] Ignacio Arroyo-Fernández, Carlos-Francisco Méndez-Cruz, Gerardo Sierra, Juan-Manuel Torres-Moreno, and Grigori Sidorov. Unsupervised sentence representations as word information series: Revisiting tf–idf. *Computer Speech & Language*, 56:107–129, 2019. ISSN 0885-2308. doi: https://doi.org/10.1016/j.csl.2019.01.005. URL https://www.sciencedirect.com/science/article/pii/S0885230817302887.

[11] Tara Athan, Harold Boley, Guido Governatori, Monica Palmirani, Adrian Paschke, and Adam Z Wyner. Oasis legalruleml. In *ICAIL*, volume 13, pages 3–12, 2013.

[12] J.L. Austin, J.O. Urmson, and M. Sbisà. *How to Do Things with Words*. William James lectures. Clarendon Press, 1975. ISBN 9780198245537. URL https://books.google.it/books?id=XnRkQSTUpmgC.

[13] Emgad H. Bachoore and Hans L. Bodlaender. Weighted treewidth algorithmic techniques and results. In Takeshi Tokuyama, editor, *Algorithms and Computation, 18th International Symposium, ISAAC 2007, Sendai, Japan, December 17-19, 2007, Proceedings*, volume 4835 of *Lecture Notes in Computer Science*, pages 893–903. Springer, 2007. doi: 10.1007/978-3-540-77120-3_77. URL https://doi.org/10.1007/978-3-540-77120-3_77.

[14] Bram Bakker. Reinforcement learning with long short-term memory. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in*

Neural Information Processing Systems, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/file/a38b16173474ba8b1a95bcbc30d3b8a5-Paper.pdf.

[15] Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. Abstract meaning representation for sembanking. In Stefanie Dipper, Maria Liakata, and Antonio Pareja-Lora, editors, *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse, LAW-ID@ACL 2013, August 8-9, 2013, Sofia, Bulgaria*, pages 178–186. The Association for Computer Linguistics, 2013. URL https://aclanthology.org/W13-2322/.

[16] Petr Baudis and Jan Sedivý. Modeling of the question answering task in the yodaqa system. In Josiane Mothe, Jacques Savoy, Jaap Kamps, Karen Pinel-Sauvagnat, Gareth J. F. Jones, Eric SanJuan, Linda Cappellato, and Nicola Ferro, editors, *Experimental IR Meets Multilinguality, Multimodality, and Interaction - 6th International Conference of the CLEF Association, CLEF 2015, Toulouse, France, September 8-11, 2015, Proceedings*, volume 9283 of *Lecture Notes in Computer Science*, pages 222–228. Springer, 2015. doi: 10.1007/978-3-319-24027-5_20. URL https://doi.org/10.1007/978-3-319-24027-5_20.

[17] Brian Beckage, Stuart Kauffman, Louis J. Gross, Asim Zia, and Christopher Koliba. *More Complex Complexity: Exploring the Nature of Computational Irreducibility across Physical, Biological, and Human Social Systems*, pages 79–88. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-35482-3. doi: 10.1007/978-3-642-35482-3_7. URL https://doi.org/10.1007/978-3-642-35482-3_7.

[18] Joel P. Beier and Martina A. Rau. Embodied learning with physical and virtual manipulatives in an intelligent tutor for chemistry. In Maria Mercedes T. Rodrigo, Noburu Matsuda, Alexandra I. Cristea, and Vania Dimitrova, editors, *Artificial Intelligence in Education - 23rd International Conference, AIED 2022, Durham, UK, July 27-31, 2022, Proceedings, Part I*, volume 13355 of *Lecture Notes in Computer Science*, pages 103–114. Springer, 2022. doi: 10.1007/978-3-031-11644-5_9. URL https://doi.org/10.1007/978-3-031-11644-5_9.

[19] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,*

EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1533–1544. ACL, 2013. URL https://aclanthology.org/D13-1160/.

[20] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe Model-based Reinforcement Learning with Stability Guarantees. In I Guyon, U V Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/766ebcd59621e305170616ba3d3dac32-Paper.pdf.

[21] Leema Kuhn Berland and Brian J Reiser. Making sense of argumentation and explanation. *Science Education*, 93(1):26–55, 2009.

[22] Adrien Bibal, Michael Lognoul, Alexandre de Streel, and Benoît Frénay. Legal requirements on explainability in machine learning. *Artif. Intell. Law*, 29(2):149–169, 2021. doi: 10.1007/s10506-020-09270-4. URL https://doi.org/10.1007/s10506-020-09270-4.

[23] Jan Blumenkamp, Steven D. Morad, Jennifer Gielis, Qingbiao Li, and Amanda Prorok. A framework for real-world multi-robot systems running decentralized gnn-based policies. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 8772–8778. IEEE, 2022. doi: 10.1109/ICRA46639.2022.9811744. URL https://doi.org/10.1109/ICRA46639.2022.9811744.

[24] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information processing letters*, 24(6):377–380, 1987.

[25] P.S. Bokare and A.K. Maurya. Acceleration-deceleration behaviour of various vehicle types. *Transportation Research Procedia*, 25:4733–4749, 2017. ISSN 2352-1465. doi: https://doi.org/10.1016/j.trpro.2017.05.486. URL https://www.sciencedirect.com/science/article/pii/S2352146517307937. World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.

[26] Simone Borsci, Stefano Federici, Silvia Bacci, Michela Gnaldi, and Francesco Bartolucci. Assessing user satisfaction in the era of user experience: Comparison of the sus, umux, and UMUX-LITE as a function of

product experience. *Int. J. Hum. Comput. Interact.*, 31(8):484–495, 2015. doi: 10.1080/10447318.2015.1064648. URL https://doi.org/10.1080/10447318.2015.1064648.

[27] Johan Bos. Expressive power of abstract meaning representations. *Comput. Linguistics*, 42(3):527–535, 2016. doi: 10.1162/COLI_a_00257. URL https://doi.org/10.1162/COLI_a_00257.

[28] Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 632–642. The Association for Computational Linguistics, 2015. doi: 10.18653/v1/d15-1075. URL https://doi.org/10.18653/v1/d15-1075.

[29] L. Karl Branting. Building explanations from rules and structured cases. *International Journal of Man-Machine Studies*, 34(6):797–837, 1991. ISSN 0020-7373. doi: https://doi.org/10.1016/0020-7373(91)90012-V. URL https://www.sciencedirect.com/science/article/pii/002073739190012V.

[30] A. Bretto. *Hypergraph Theory: An Introduction*. Mathematical Engineering. Springer International Publishing, 2013. ISBN 9783319000800. URL https://books.google.co.uk/books?id=1b5DAAAAQBAJ.

[31] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[32] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=F72ximsx7C1.

[33] Sylvain Bromberger. Why-questions. In Robert G. Colodny, editor, *Mind and Cosmos – Essays in Contemporary Science and Philosophy*, pages 86–111. University of Pittsburgh Press, 1966.

[34] John Brooke. Sus: a retrospective. *Journal of usability studies*, 8(2):29–40, 2013.

[35] Teresa Kissane Brostoff and Ann Sinsheimer. *United States Legal Language and Culture: An Introduction to the U.S. Common Law System*. Oxford

University Press USA, 2013. ISBN 9780199895458. URL https://books.google.it/books?id=SVsGAQAAQBAJ.

[36] Georg Brun. Explication as a method of conceptual re-engineering. *Erkenntnis*, 81(6):1211–1241, 2016. doi: 10.1007/s10670-015-9791-5.

[37] Zana Buçinca, Phoebe Lin, Krzysztof Z. Gajos, and Elena L. Glassman. Proxy tasks and subjective measures can be misleading in evaluating explainable AI systems. In Fabio Paternò, Nuria Oliver, Cristina Conati, Lucio Davide Spano, and Nava Tintarev, editors, *IUI '20: 25th International Conference on Intelligent User Interfaces, Cagliari, Italy, March 17-20, 2020*, pages 454–464. ACM, 2020. doi: 10.1145/3377325.3377498. URL https://doi.org/10.1145/3377325.3377498.

[38] Elena Cabrio, Sara Tonelli, and Serena Villata. From discourse analysis to argumentation schemes and back: Relations and differences. In João Leite, Tran Cao Son, Paolo Torroni, Leon van der Torre, and Stefan Woltran, editors, *Computational Logic in Multi-Agent Systems - 14th International Workshop, CLIMA XIV, Corunna, Spain, September 16-18, 2013. Proceedings*, volume 8143 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2013. URL https://doi.org/10.1007/978-3-642-40624-9_1.

[39] John T. Cacioppo and Richard E. Petty. The need for cognition. *Journal of Personality and Social Psychology*, 42(1):116–131, 1982. doi: 10.1037/0022-3514.42.1.116.

[40] Rachel Van Campenhout, Nick Brown, Bill Jerome, Jeffrey S. Dittel, and Benny G. Johnson. Toward effective courseware at scale: Investigating automatically generated questions as formative practice. In Christoph Meinel, Mar Pérez-Sanagustín, Marcus Specht, and Amy Ogan, editors, *L@S'21: Eighth ACM Conference on Learning @ Scale, Virtual Event, Germany, June 22-25, 2021*, pages 295–298. ACM, 2021. doi: 10.1145/3430895.3460162. URL https://doi.org/10.1145/3430895.3460162.

[41] Rachel Van Campenhout, Jeffrey S. Dittel, Bill Jerome, and Benny G. Johnson. Transforming textbooks into learning by doing environments: An evaluation of textbook-based automatic question generation. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, and Andrew S. Lan, editors, *Proceedings of the Third International Workshop on Inteligent Textbooks 2021 Co-located with 22nd International Conference on Artificial Intelligence in Education (AIED 2021), Online, June 15, 2021*, volume 2895

of *CEUR Workshop Proceedings*, pages 60–73. CEUR-WS.org, 2021. URL http://ceur-ws.org/Vol-2895/paper06.pdf.

[42] Rudolf Carnap and Paul A Schilpp. *The Philosophy of Rudolf Carnap*. Cambridge University Press Cambridge, 1963.

[43] Corinne Cath, Sandra Wachter, Brent D. Mittelstadt, Mariarosaria Taddeo, and Luciano Floridi. Artificial intelligence and the 'good society': the us, eu, and UK approach. *Sci. Eng. Ethics*, 24(2):505–528, 2018. doi: 10.1007/s11948-017-9901-7. URL https://doi.org/10.1007/s11948-017-9901-7.

[44] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2029. URL https://aclanthology.org/D18-2029.

[45] Isaac Alpizar Chacon, Jordan Barria-Pineda, Kamil Akhuseyinoglu, Sergey A. Sosnovsky, and Peter Brusilovsky. Integrating textbooks with smart interactive content for learning programming. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, and Andrew S. Lan, editors, *Proceedings of the Third International Workshop on Inteligent Textbooks 2021 Co-located with 22nd International Conference on Artificial Intelligence in Education (AIED 2021), Online, June 15, 2021*, volume 2895 of *CEUR Workshop Proceedings*, pages 4–18. CEUR-WS.org, 2021. URL http://ceur-ws.org/Vol-2895/paper11.pdf.

[46] Ilias Chalkidis and Dimitrios Kampas. Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artif. Intell. Law*, 27(2):171–198, 2019. URL https://doi.org/10.1007/s10506-018-9238-9.

[47] Danqi Chen and Wen-tau Yih. Open-domain question answering. In Agata Savary and Yue Zhang, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, ACL 2020, Online, July 5, 2020*, pages 34–37. Association for Computational Linguistics, 2020. URL https://doi.org/10.18653/v1/2020.acl-tutorials.8.

[48] Dongmei Chen, Sheng Zhang, Xin Zhang, and Kaijing Yang. Cross-lingual passage re-ranking with alignment augmented multilingual BERT. *IEEE Access*, 8:213232–213243, 2020. doi: 10.1109/ACCESS.2020.3041605. URL https://doi.org/10.1109/ACCESS.2020.3041605.

[49] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM, 2016. doi: 10.1145/2939672.2939785. URL https://doi.org/10.1145/2939672.2939785.

[50] Yinlam Chow, Ofir Nachum, Edgar A. Duéñez-Guzmán, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 8103–8112, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/4fe5149039b52765bde64beb9f674940-Abstract.html.

[51] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. *CoRR*, abs/2204.02311, 2022. doi: 10.48550/arXiv.2204.02311. URL https://doi.org/10.48550/arXiv.2204.02311.

[52] Paul D. Clough and Mark Sanderson. Evaluating the performance of information retrieval systems using test collections. *Inf. Res.*, 18(2), 2013. URL http://www.informationr.net/ir/18-2/paper582.html.

[53] European Commission. White paper on artificial intelligence: A european approach to excellence and trust, 2020. URL https://ec.europa.eu/info/publications/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en.

[54] Mehul Damani, Zhiyao Luo, Emerson Wenzel, and Guillaume Sartoretti. Primal$_2$: Pathfinding via reinforcement and imitation multi-agent learning - lifelong. *IEEE Robotics and Automation Letters*, 6(2):2666–2673, April 2021. ISSN 2377-3766. doi: 10.1109/LRA.2021.3062803.

[55] Domenico Dato, Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, and Nicola Tonellotto. The istella22 dataset: Bridging traditional and neural learning to rank evaluation. In Enrique Amigó, Pablo Castells, Julio Gonzalo, Ben Carterette, J. Shane Culpepper, and Gabriella Kazai, editors, *SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, July 11 - 15, 2022*, pages 3099–3107. ACM, 2022. doi: 10.1145/3477495.3531740. URL https://doi.org/10.1145/3477495.3531740.

[56] Gabriel Lins de Holanda Coelho, Paul H. P. Hanel, and Lukas J. Wolf. The very efficient assessment of need for cognition: Developing a six-item version. *Assessment*, 27(8):1870–1885, 2020. doi: 10.1177/1073191118793208. URL https://doi.org/10.1177/1073191118793208. PMID: 30095000.

[57] Boris Delaunay. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.

[58] Nick Deligiannis, Dionysis Panagiotopoulos, Panagiotis Patsilinakos, Chrysanthi N. Raftopoulou, and Antonios Symvonis. Interactive and personalized activity ebooks for learning to read: The iread case. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, Rakesh Agrawal, and Andrew S. Lan, editors, *Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019*, volume 2384 of *CEUR Workshop Proceedings*, pages 57–69. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2384/paper06.pdf.

[59] Robert Detrano, Andras Janosi, Walter Steinbrunn, Matthias Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern H. Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *The American Journal of Cardiology*, 64(5):304–310, 1989. ISSN 0002-9149. doi: https://doi.org/10.1016/0002-9149(89)90524-9. URL https://www.sciencedirect.com/science/article/pii/0002914989905249.

[60] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

[61] EPRS DG. Understanding algorithmic decision-making: Opportunities and challenges, 2019. URL https://www.europarl.europa.eu/thinktank/en/document/EPRS_STU(2019)624261.

[62] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Pai-Shun Ting, Karthikeyan Shanmugam, and Payel Das. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 590–601, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/c5ff2543b53f4cc0ad3819a36752467b-Abstract.html.

[63] Jürgen Dieber and Sabrina Kirrane. A novel model usability evaluation framework (muse) for explainable artificial intelligence. *Inf. Fusion*, 81:143–153, 2022. doi: 10.1016/j.inffus.2021.11.017. URL https://doi.org/10.1016/j.inffus.2021.11.017.

[64] Thomas G. Dietterich and Nicholas S. Flann. Explanation-based learning and reinforcement learning: A unified view. *Mach. Learn.*, 28(2-3):169–210, 1997. doi: 10.1023/A:1007355226281. URL https://doi.org/10.1023/A:1007355226281.

[65] Robert DiPietro and Gregory D. Hager. Chapter 21 - deep learning: Rnns and lstm. In S. Kevin Zhou, Daniel Rueckert, and Gabor Fichtinger, editors, *Handbook of Medical Image Computing and Computer Assisted Intervention*, The Elsevier and MICCAI Society Book Series, pages 503–519. Academic Press, 2020. ISBN 978-0-12-816176-0. doi: https://doi.org/10.1016/B978-0-12-816176-0.00026-0. URL https://www.sciencedirect.com/science/article/pii/B9780128161760000260.

[66] Igor Douven. Peter achinstein: Evidence, explanation, and realism: Essays in philosophy of science. *Science & Education*, 21(4):597–601, 2012. doi: 10.1007/s11191-011-9405-9. URL https://doi.org/10.1007/s11191-011-9405-9.

[67] Jingcheng Du, Qingyu Chen, Yifan Peng, Yang Xiang, Cui Tao, and Zhiyong Lu. Ml-net: multi-label classification of biomedical texts with deep neural networks. *J Am Med Inform Assoc*, 26(11):1279–1285, Nov 2019. ISSN 1527-974X (Electronic); 1067-5027 (Print); 1067-5027 (Linking). doi: 10.1093/jamia/ocz085.

[68] Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott C. Deerwester, and Richard A. Harshman. Using latent semantic analysis to improve access to textual information. In J. J. O'Hare, editor, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 1988, Washington, D.C., USA, May 15-19, 1988*, pages 281–285. ACM, 1988. doi: 10.1145/57167.57214. URL https://doi.org/10.1145/57167.57214.

[69] Martin Ebers. Regulating explainable ai in the european union. an overview of the current legal framework (s). In Liane Colonna and Stanley Greenstein, editors, *Nordic Yearbook of Law and Informatics 2020: Law in the Era of Artificial Intelligence*. SSRN, 2021. doi: 10.2139/ssrn.3901732. URL https://ssrn.com/abstract=3901732.

[70] Barbara Ericson. An analysis of interactive feature use in two ebooks. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, Rakesh Agrawal, and Andrew S. Lan, editors, *Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019*, volume 2384 of *CEUR Workshop Proceedings*, pages 4–17. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2384/paper01.pdf.

[71] Daniel B Fambro, Rodger J Koppa, Dale L Picha, and Kay Fitzpatrick. Driver Braking Performance in Stopping Sight Distance Situations. *Transportation Research Record*, 1701(1):9–16, 1 2000. ISSN 0361-1981. doi: 10.3141/1701-02. URL https://doi.org/10.3141/1701-02.

[72] John Rupert Firth and Philological Society (Great Britain). *Studies in Linguistic Analysis*. Publications of the Philological Society. Blackwell, 1957. ISBN 9780631113003. URL https://books.google.it/books?id=JWktAAAAMAAJ.

[73] Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. Large-scale QA-SRL parsing. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2051–2060. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1191. URL https://aclanthology.org/P18-1191/.

[74] Luciano Floridi, Josh Cowls, Monica Beltrametti, Raja Chatila, Patrice Chazerand, Virginia Dignum, Christoph Luetge, Robert Madelin, Ugo Pagallo, Francesca Rossi, Burkhard Schafer, Peggy Valcke, and Effy Vayena. Ai4people - an ethical framework for a good AI society: Opportunities, risks, principles, and recommendations. *Minds Mach.*, 28(4):689–707, 2018. doi: 10.1007/s11023-018-9482-5. URL https://doi.org/10.1007/s11023-018-9482-5.

[75] Jakob N. Foerster, Nantas Nardelli, Gregory Farquhar, Triantafyllos Afouras, Philip H. S. Torr, Pushmeet Kohli, and Shimon Whiteson. Stabilising experience replay for deep multi-agent reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1146–1155. PMLR, 2017. URL http://proceedings.mlr.press/v70/foerster17b.html.

[76] International Organization for Standardization. *Ergonomics of human-system interaction: Part 210: Human-centred design for interactive systems*. ISO, 2010.

[77] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July*

10-15, 2018, volume 80 of *Proceedings of Machine Learning Research*, pages 1582–1591. PMLR, 2018. URL http://proceedings.mlr.press/v80/fujimoto18a.html.

[78] Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.*, 16:1437–1480, 2015. doi: 10.5555/2789272.2886795. URL https://dl.acm.org/doi/10.5555/2789272.2886795.

[79] Jennifer Gielis, Ajay Shankar, and Amanda Prorok. A critical review of communications in multi-robot systems. *Current Robotics Reports*, 2022. doi: 10.1007/s43154-022-00090-9. URL https://doi.org/10.1007/s43154-022-00090-9.

[80] Leilani H. Gilpin, David Bau, Ben Z. Yuan, Ayesha Bajwa, Michael A. Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In Francesco Bonchi, Foster J. Provost, Tina Eliassi-Rad, Wei Wang, Ciro Cattuto, and Rayid Ghani, editors, *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*, pages 80–89. IEEE, 2018. doi: 10.1109/DSAA.2018.00018. URL https://doi.org/10.1109/DSAA.2018.00018.

[81] Georg Gottlob, Gianluigi Greco, Nicola Leone, and Francesco Scarcello. Hypertree decompositions: Questions and answers. In Tova Milo and Wang-Chiew Tan, editors, *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 57–74. ACM, 2016. doi: 10.1145/2902251.2902309. URL https://doi.org/10.1145/2902251.2902309.

[82] John C Gower and Gavin JS Ross. Minimum spanning trees and single linkage cluster analysis. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 18(1):54–64, 1969.

[83] Mandy Guo, Yinfei Yang, Daniel Cer, Qinlan Shen, and Noah Constant. MultiReQA: A cross-domain evaluation forRetrieval question answering models. In *Proceedings of the Second Workshop on Domain Adaptation for NLP*, pages 94–104, Kyiv, Ukraine, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.adaptnlp-1.10.

[84] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

[85] Philipp Hacker and Jan-Hendrik Passoth. Varieties of AI explanations under the law. from the GDPR to the aia, and beyond. In Andreas Holzinger, Randy Goebel, Ruth Fong, Taesup Moon, Klaus-Robert Müller, and Wojciech Samek, editors, *xxAI - Beyond Explainable AI - International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers*, volume 13200 of *Lecture Notes in Computer Science*, pages 343–373. Springer, 2020. doi: 10.1007/978-3-031-04083-2_17. URL https://doi.org/10.1007/978-3-031-04083-2_17.

[86] Zellig S. Harris. *Distributional Structure*, pages 3–22. Springer Netherlands, Dordrecht, 1981. ISBN 978-94-009-8467-7. doi: 10.1007/978-94-009-8467-7_1. URL https://doi.org/10.1007/978-94-009-8467-7_1.

[87] Luheng He, Mike Lewis, and Luke Zettlemoyer. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In Lluís Màrquez, Chris Callison-Burch, Jian Su, Daniele Pighin, and Yuval Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 643–653. The Association for Computational Linguistics, 2015. doi: 10.18653/v1/d15-1076. URL https://doi.org/10.18653/v1/d15-1076.

[88] Carl G. Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of Science*, 15(2):135–175, 1948. doi: 10.1086/286983.

[89] Denis J. Hilton. Mental models and causal explanation: Judgements of probable cause and explanatory relevance. *Thinking & Reasoning*, 2(4): 273–308, 1996. doi: 10.1080/135467896394447. URL https://doi.org/10.1080/135467896394447.

[90] Robert R. Hoffman, Shane T. Mueller, Gary Klein, and Jordan Litman. Metrics for explainable AI: challenges and prospects. *CoRR*, abs/1812.04608, 2018. URL http://arxiv.org/abs/1812.04608.

[91] J.H. Holland, K.J. Holyoak, R.E. Nisbett, and P.R. Thagard. *Induction: Processes of Inference, Learning, and Discovery*. Bradford books. MIT Press, 1986. ISBN 9780262580960. URL https://books.google.it/books?id=Z6EFBaLApE8C.

[92] Andreas Holzinger, André M. Carrington, and Heimo Müller. Measuring the quality of explanations: The system causability scale (SCS). *Künstliche Intell.*, 34(2):193–198, 2020. doi: 10.1007/s13218-020-00636-z. URL https://doi.org/10.1007/s13218-020-00636-z.

[93] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 328–339. Association for Computational Linguistics, 2018. doi: 10.18653/v1/P18-1031. URL https://aclanthology.org/P18-1031/.

[94] Zhen Huang, Shiyi Xu, Minghao Hu, Xinyi Wang, Jinyan Qiu, Yongquan Fu, Yuncai Zhao, Yuxing Peng, and Changjian Wang. Recent trends in deep learning based open-domain textual question answering systems. *IEEE Access*, 8:94341–94356, 2020. doi: 10.1109/ACCESS.2020.2988903. URL https://doi.org/10.1109/ACCESS.2020.2988903.

[95] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50 (2):21:1–21:35, 2017. doi: 10.1145/3054912. URL https://doi.org/10.1145/3054912.

[96] IBM. Ai explainability 360 - demo. https://aix360.mybluemix.net/explanation_cust, 2019. Online; accessed 29-Mar-2020.

[97] ICO. Project explain interim report. https://ico.org.uk/about-the-ico/research-and-reports/project-explain-interim-report/, 2019. Online; accessed 05-Jan-2020.

[98] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,*

(AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pages 3302–3309. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16054.

[99] Peter Jansen, Niranjan Balasubramanian, Mihai Surdeanu, and Peter Clark. What's in an explanation? characterizing knowledge and inference requirements for elementary science exams. In Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad, editors, *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 2956–2965. ACL, 2016. URL https://aclanthology.org/C16-1278/.

[100] Weina Jin, Sheelagh Carpendale, Ghassan Hamarneh, and Diane Gromala. Bridging ai developers and end users: an end-user-centred explainable ai taxonomy and visual vocabularies. In *Proceedings of the IEEE Visualization, Vancouver, BC, Canada*, pages 20–25, 2019.

[101] Samuel G.B. Johnson, J.J. Valenti, and Frank C. Keil. Simplicity and complexity preferences in causal explanation: An opponent heuristic account. *Cognitive Psychology*, 113:101222, 2019. ISSN 0010-0285. doi: https://doi.org/10.1016/j.cogpsych.2019.05.004. URL https://www.sciencedirect.com/science/article/pii/S0010028517300579.

[102] Karen Spärck Jones. A statistical interpretation of term specificity and its application in retrieval. *J. Documentation*, 60(5):493–502, 2004. doi: 10.1108/00220410410560573. URL https://doi.org/10.1108/00220410410560573.

[103] Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics, 2017. doi: 10.18653/v1/P17-1147. URL https://doi.org/10.18653/v1/P17-1147.

[104] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for

atari. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

[105] Margot E. Kaminski. The right to explanation, explained. *Berkeley Technology Law Journal*, 34(1):189–218, 2019. URL https://heinonline.org/HOL/P?h=hein.journals/berktech34&i=202.

[106] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick S. H. Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

[107] Judy Kay and Bob Kummerfeld. Scaffolded, scrutable open learner model (SOLM) as a foundation for personalised e-textbooks. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, Rakesh Agrawal, and Andrew S. Lan, editors, *Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019*, volume 2384 of *CEUR Workshop Proceedings*, pages 38–43. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2384/paper04.pdf.

[108] Judy Kay and Bob Kummerfeld. PUMPT: an e-textbook platform based on a personal user model for learning (short paper). In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, and Andrew S. Lan, editors, *Proceedings of the Third International Workshop on Inteligent Textbooks 2021 Co-located with 22nd International Conference on Artificial Intelligence in Education (AIED 2021), Online, June 15, 2021*, volume 2895 of *CEUR Workshop Proceedings*, pages 27–34. CEUR-WS.org, 2021. URL http://ceur-ws.org/Vol-2895/paper12.pdf.

[109] B. Ravi Kiran, Ibrahim Sobh, Victor Talpaert, Patrick Mannion, Ahmad A. Al Sallab, Senthil Kumar Yogamani, and Patrick Pérez. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.*, 23(6):4909–4926, 2022. doi: 10.1109/TITS.2021.3054625. URL https://doi.org/10.1109/TITS.2021.3054625.

[110] Ryan Kiros, Yukun Zhu, Russ R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought

vectors. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf.

[111] Ben Kluga, Manohar Sai Jasti, Virginia Naples, and Reva Freedman. Adding intelligence to a textbook for human anatomy with a causal concept map based ITS. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, Rakesh Agrawal, and Andrew S. Lan, editors, *Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019*, volume 2384 of *CEUR Workshop Proceedings*, pages 124–134. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2384/paper13.pdf.

[112] Kamran Kowsari, Mojtaba Heidarysafa, Donald E. Brown, Kiana Jafari Meimandi, and Laura E. Barnes. Rmdl: Random multimodel deep learning for classification. In *Proceedings of the 2nd International Conference on Information System and Data Mining*, ICISDM '18, page 19–28, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450363549. doi: 10.1145/3206098.3206111. URL https://doi.org/10.1145/3206098.3206111.

[113] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Trans. Assoc. Comput. Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://doi.org/10.1162/tacl_a_00276.

[114] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Interpretable & explorable approximations of black box models. *CoRR*, abs/1707.01154, 2017. URL http://arxiv.org/abs/1707.01154.

[115] Ho-Pun Lam and Guido Governatori. The making of spindle. In Guido Governatori, John Hall, and Adrian Paschke, editors, *Rule Interchange and Applications, International Symposium, RuleML 2009, Las Vegas, Nevada, USA, November 5-7, 2009. Proceedings*, volume 5858 of *Lecture Notes in Computer Science*, pages 315–322. Springer, 2009. URL https://doi.org/10.1007/978-3-642-04985-9_29.

[116] Irene Langkilde and Kevin Knight. Generation that exploits corpus-based statistical knowledge. In Christian Boitet and Pete Whitelock, editors, *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, COLING-ACL '98, August 10-14, 1998, Université de Montréal, Montréal, Quebec, Canada. Proceedings of the Conference*, pages 704–710. Morgan Kaufmann Publishers / ACL, 1998. doi: 10.3115/980845.980963. URL https://aclanthology.org/P98-1116/.

[117] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page II–1188–II–1196. JMLR.org, 2014.

[118] Yunjiao Lei, Dayong Ye, Sheng Shen, Yulei Sui, Tianqing Zhu, and Wanlei Zhou. New challenges in reinforcement learning: a survey of security and privacy. *Artificial Intelligence Review*, 2022. doi: 10.1007/s10462-022-10348-5. URL https://doi.org/10.1007/s10462-022-10348-5.

[119] James R. Lewis. Measuring perceived usability: The csuq, sus, and UMUX. *Int. J. Hum. Comput. Interact.*, 34(12):1148–1156, 2018. doi: 10.1080/10447318.2017.1418805. URL https://doi.org/10.1080/10447318.2017.1418805.

[120] Changjian Li and Krzysztof Czarnecki. Urban driving with multi-objective deep reinforcement learning. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor, editors, *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 359–367. International Foundation for Autonomous Agents and Multiagent Systems, 2019. URL http://dl.acm.org/citation.cfm?id=3331714.

[121] Gen Li, Yuxin Chen, Yuejie Chi, Yuantao Gu, and Yuting Wei. Sample-efficient reinforcement learning is feasible for linearly realizable mdps with limited revisiting. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 16671–16685, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/8b5700012be65c9da25f49408d959ca0-Abstract.html.

[122] Qingbiao Li, Weizhe Lin, Zhe Liu, and Amanda Prorok. Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robotics Autom. Lett.*, 6(3):5533–5540, 2021. doi: 10.1109/LRA.2021.3077863. URL https://doi.org/10.1109/LRA.2021.3077863.

[123] Q. Vera Liao and Kush R. Varshney. Human-centered explainable AI (XAI): from algorithms to user experiences. *CoRR*, abs/2110.10790, 2021. URL https://arxiv.org/abs/2110.10790.

[124] Q. Vera Liao, Daniel M. Gruen, and Sarah Miller. Questioning the AI: informing design practices for explainable AI user experiences. In Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik, editors, *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, pages 1–15. ACM, 2020. doi: 10.1145/3313831.3376590. URL https://doi.org/10.1145/3313831.3376590.

[125] Brian Y. Lim, Anind K. Dey, and Daniel Avrahami. *Why and why not* explanations improve the intelligibility of context-aware intelligent systems. In Dan R. Olsen Jr., Richard B. Arthur, Ken Hinckley, Meredith Ringel Morris, Scott E. Hudson, and Saul Greenberg, editors, *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 2119–2128. ACM, 2009. doi: 10.1145/1518701.1519023. URL https://doi.org/10.1145/1518701.1519023.

[126] Scott M. Lundberg, Gabriel G. Erion, Hugh Chen, Alex J. DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.*, 2(1):56–67, 2020. doi: 10.1038/s42256-019-0138-9. URL https://doi.org/10.1038/s42256-019-0138-9.

[127] Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. A grounded interaction protocol for explainable artificial intelligence. In Edith Elkind, Manuela Veloso, Noa Agmon, and Matthew E. Taylor, editors, *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019*, pages 1033–1041. International Foundation for Autonomous Agents and Multiagent Systems, 2019. URL http://dl.acm.org/citation.cfm?id=3331801.

[128] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947.

[129] Robert Martin. *Agile software development: principles, patterns, and practices*. Prentice Hall, 2002. ISBN 978-0-135-97444-5.

[130] Noboru Matsuda and Machi Shimmei. PASTEL: evidence-based learning engineering method to create intelligent online textbook at scale. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, Rakesh Agrawal, and Andrew S. Lan, editors, *Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019*, volume 2384 of *CEUR Workshop Proceedings*, pages 70–80. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2384/paper07.pdf.

[131] G. Randolph Mayes. Theories of explanation, 2001. URL https://iep.utm.edu/explanat/.

[132] Bogdan Mazoure, Thang Doan, Audrey Durand, Joelle Pineau, and R. Devon Hjelm. Leveraging exploration in off-policy algorithms via normalizing flows. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *3rd Annual Conference on Robot Learning, CoRL 2019, Osaka, Japan, October 30 - November 1, 2019, Proceedings*, volume 100 of *Proceedings of Machine Learning Research*, pages 430–444. PMLR, 2019. URL http://proceedings.mlr.press/v100/mazoure20a.html.

[133] Scott McDonald and Michael Ramscar. Testing the distributioanl hypothesis: The influence of context on judgements of semantic similarity. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 23, 2001.

[134] Fengchun Miao, Wayne Holmes, Ronghuai Huang, and Hui Zhang. *AI and education: A guidance for policymakers*. UNESCO Publishing, 2021.

[135] Julian Michael, Gabriel Stanovsky, Luheng He, Ido Dagan, and Luke Zettlemoyer. Crowdsourcing question-answer meaning representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 560–568. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-2089. URL https://doi.org/10.18653/v1/n18-2089.

[136] Nathan Michael, Michael M. Zavlanos, Vijay Kumar, and George J. Pappas. Distributed multi-robot task assignment and formation control. In *2008 IEEE International Conference on Robotics and Automation, ICRA 2008, May 19-23, 2008, Pasadena, California, USA*, pages 128–133. IEEE, 2008. doi: 10.1109/ROBOT.2008.4543197. URL https://doi.org/10.1109/ROBOT.2008.4543197.

[137] Martijn Millecamp, Nyi Nyi Htun, Cristina Conati, and Katrien Verbert. To explain or not to explain: the effects of personal characteristics when explaining music recommendations. In Wai-Tat Fu, Shimei Pan, Oliver Brdiczka, Polo Chau, and Gaelle Calvary, editors, *Proceedings of the 24th International Conference on Intelligent User Interfaces, IUI 2019, Marina del Ray, CA, USA, March 17-20, 2019*, pages 397–407. ACM, 2019. doi: 10.1145/3301275.3302313. URL https://doi.org/10.1145/3301275.3302313.

[138] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artif. Intell.*, 267:1–38, 2019. doi: 10.1016/j.artint.2018.07.007. URL https://doi.org/10.1016/j.artint.2018.07.007.

[139] Eleni Miltsakaki, Rashmi Prasad, Aravind K. Joshi, and Bonnie L. Webber. The penn discourse treebank. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004, May 26-28, 2004, Lisbon, Portugal*. European Language Resources Association, 2004. URL http://www.lrec-conf.org/proceedings/lrec2004/summaries/618.htm.

[140] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. ISSN 1476-4687. doi: 10.1038/nature14236. URL https://doi.org/10.1038/nature14236.

[141] Mostafa Mohammed and Clifford A. Shaffer. Increasing student interaction with an etextbook using programmed instruction (short paper). In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, and Andrew S. Lan, editors, *Proceedings of the Third International Workshop on Inteligent Textbooks 2021 Co-located with 22nd International Conference on Artificial Intelligence in Education (AIED 2021), Online, June 15, 2021*, volume 2895

of *CEUR Workshop Proceedings*, pages 40–44. CEUR-WS.org, 2021. URL http://ceur-ws.org/Vol-2895/paper14.pdf.

[142] Sina Mohseni, Jeremy E. Block, and Eric D. Ragan. Quantitative evaluation of machine learning explanations: A human-grounded benchmark. In Tracy Hammond, Katrien Verbert, Dennis Parra, Bart P. Knijnenburg, John O'Donovan, and Paul Teale, editors, *IUI '21: 26th International Conference on Intelligent User Interfaces, College Station, TX, USA, April 13-17, 2021*, pages 22–31. ACM, 2021. doi: 10.1145/3397481.3450689. URL https://doi.org/10.1145/3397481.3450689.

[143] An-phi Nguyen and María Rodríguez Martínez. On quantitative aspects of model interpretability. *CoRR*, abs/2007.07584, 2020. URL https://arxiv.org/abs/2007.07584.

[144] Isack Thomas Nicholaus and Dae-Ki Kang. Robust experience replay sampling for multi-agent reinforcement learning. *Pattern Recognit. Lett.*, 155:135–142, 2022. doi: 10.1016/j.patrec.2021.11.006. URL https://doi.org/10.1016/j.patrec.2021.11.006.

[145] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. In Beth Adelson, Susan T. Dumais, and Judith S. Olson, editors, *Conference on Human Factors in Computing Systems, CHI 1994, Boston, Massachusetts, USA, April 24-28, 1994, Proceedings*, pages 152–158. ACM, 1994. doi: 10.1145/191666.191729. URL https://doi.org/10.1145/191666.191729.

[146] Donald Nute and Katrin Erk. Defeasible logic graphs: I. theory. *Decis. Support Syst.*, 22(3):277–293, 1998. doi: 10.1016/S0167-9236(97)00063-8. URL https://doi.org/10.1016/S0167-9236(97)00063-8.

[147] High-Level Expert Group on Artificial Intelligence. Ethics guidelines for trustworthy ai, 2019. URL https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai.

[148] High-Level Expert Group on Artificial Intelligence. Policy and investment recommendations, 2019. URL https://digital-strategy.ec.europa.eu/en/library/policy-and-investment-recommendations-trustworthy-artificial-intelligence.

[149] Ugo Pagallo. Algoritmi e conoscibilità. *Rivista di filosofia del diritto*, 9(1):93–106, 2020. doi: 10.4477/97022. URL https://www.rivisteweb.it/doi/10.4477/97022.

[150] Stefan Palan and Christian Schitter. Prolific.ac—a subject pool for online experiments. *Journal of Behavioral and Experimental Finance*, 17:22–27, 2018. ISSN 2214-6350. doi: https://doi.org/10.1016/j.jbef.2017.12.004. URL https://www.sciencedirect.com/science/article/pii/S2214635017300989.

[151] Monica Palmirani. Big data e conoscenza. *Rivista di filosofia del diritto*, 9 (1):73–92, 2020. doi: 10.4477/97021. URL https://www.rivisteweb.it/doi/10.4477/97021.

[152] Monica Palmirani and Guido Governatori. Modelling legal knowledge for gdpr compliance checking. In *JURIX*, pages 101–110, 2018.

[153] Steven C Pan and Timothy C Rickard. Transfer of test-enhanced learning: Meta-analytic review and synthesis. *Psychological bulletin*, 144(7):710, 2018.

[154] Teresa Phelps and Kevin Ashley. " alexa, write a memo": The promise and challenges of ai and legal writing. *Legal Writing: J. Legal Writing Inst.*, 26: 329, 2022.

[155] Justin Picard. Finding content-bearing terms using term similarities. In *EACL 1999, 9th Conference of the European Chapter of the Association for Computational Linguistics, June 8-12, 1999, University of Bergen, Bergen, Norway*, pages 241–244. The Association for Computer Linguistics, 1999. URL https://aclanthology.org/E99-1034/.

[156] Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna M. Wallach. Manipulating and measuring model interpretability. In Yoshifumi Kitamura, Aaron Quigley, Katherine Isbister, Takeo Igarashi, Pernille Bjørn, and Steven Mark Drucker, editors, *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 237:1–237:52. ACM, 2021. doi: 10.1145/3411764.3445315. URL https://doi.org/10.1145/3411764.3445315.

[157] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K. Joshi, and Bonnie L. Webber. The penn discourse treebank 2.0. In *Proceedings of the International Conference on Language Resources and Evaluation, LREC 2008, 26 May - 1 June 2008, Marrakech, Morocco*. European Language Resources Association, 2008. URL http://www.lrec-conf.org/proceedings/lrec2008/summaries/754.html.

[158] Amanda Prorok. Redundant robot assignment on graphs with uncertain edge costs. In Nikolaus Correll, Mac Schwager, and Michael W. Otte, editors, *Distributed Autonomous Robotic Systems, The 14th International Symposium, DARS 2018, Boulder, CO, USA, October 15-17, 2018*, volume 9 of *Springer Proceedings in Advanced Robotics*, pages 313–327. Springer, 2018. doi: 10.1007/978-3-030-05816-6_22. URL https://doi.org/10.1007/978-3-030-05816-6_22.

[159] Bart Pursel, Crystal M. Ramsay, Nesirag Dave, Chen Liang, and C. Lee Giles. Bbookx: Creating semi-automated textbooks to support student learning and decrease student costs. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, Rakesh Agrawal, and Andrew S. Lan, editors, *Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019*, volume 2384 of *CEUR Workshop Proceedings*, pages 81–86. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2384/paper08.pdf.

[160] Valentina Pyatkin, Ayal Klein, Reut Tsarfaty, and Ido Dagan. Qadiscourse - discourse relations as QA pairs: Representation, crowdsourcing and baselines. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 2804–2819. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.224. URL https://doi.org/10.18653/v1/2020.emnlp-main.224.

[161] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[162] Alex Raymond, Hatice Gunes, and Amanda Prorok. Culture-Based Explainable Human-Agent Deconfliction. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, pages 1107–1115, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

[163] Alex Raymond, Matthew Malencia, Guilherme Paulino-Passos, and Amanda Prorok. Agree to disagree: Subjective fairness in privacy-restricted decentralised conflict resolution. *Frontiers Robotics AI*, 9:733876, 2022.

doi: 10.3389/frobt.2022.733876. URL https://doi.org/10.3389/frobt.2022.733876.

[164] Juan Carlo Rebanal, Jordan Combitsis, Yuqi Tang, and Xiang 'Anthony' Chen. Xalgo: a design probe of explaining algorithms' internal states via question-answering. In Tracy Hammond, Katrien Verbert, Dennis Parra, Bart P. Knijnenburg, John O'Donovan, and Paul Teale, editors, *IUI '21: 26th International Conference on Intelligent User Interfaces, College Station, TX, USA, April 13-17, 2021*, pages 329–339. ACM, 2021. doi: 10.1145/3397481.3450676. URL https://doi.org/10.1145/3397481.3450676.

[165] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410. URL https://doi.org/10.18653/v1/D19-1410.

[166] Zhipeng Ren, Daoyi Dong, Huaxiong Li, and Chunlin Chen. Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning. *IEEE Trans. Neural Networks Learn. Syst.*, 29(6):2216–2226, 2018. doi: 10.1109/TNNLS.2018.2790981. URL https://doi.org/10.1109/TNNLS.2018.2790981.

[167] Mireia Ribera and Àgata Lapedriza. Can we do better explanations? A proposal of user-centered explainable AI. In Christoph Trattner, Denis Parra, and Nathalie Riche, editors, *Joint Proceedings of the ACM IUI 2019 Workshops co-located with the 24th ACM Conference on Intelligent User Interfaces (ACM IUI 2019), Los Angeles, USA, March 20, 2019*, volume 2327 of *CEUR Workshop Proceedings*, page 38. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2327/IUI19WS-ExSS2019-12.pdf.

[168] Michelle L. Rivers. Metacognition about practice testing: a review of learners'beliefs, monitoring, and control of test-enhanced learning. *Educational Psychology Review*, 33(3):823–862, 2021. doi: 10.1007/s10648-020-09578-2. URL https://doi.org/10.1007/s10648-020-09578-2.

[169] Livio Robaldo, Eleni Miltsakaki, and Jerry R. Hobbs. Refining the meaning of sense labels in PDTB: Concession. In Johan Bos and Rodolfo Delmonte, editors, *Semantics in Text Processing. STEP 2008 Conference Proceedings,*

Venice, Italy, September 22-24, 2008. Association for Computational Linguistics, 2008. URL https://aclanthology.org/W08-2217/.

[170] Jikun Rong and Nan Luan. Safe Reinforcement Learning with Policy-Guided Planning for Autonomous Driving. In *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, pages 320–326, 2020. doi: 10.1109/ICMA49215.2020.9233522.

[171] Avi Rosenfeld. Better metrics for evaluating explainable artificial intelligence. In Frank Dignum, Alessio Lomuscio, Ulle Endriss, and Ann Nowé, editors, *AAMAS '21: 20th International Conference on Autonomous Agents and Multiagent Systems, Virtual Event, United Kingdom, May 3-7, 2021*, pages 45–50. ACM, 2021. doi: 10.5555/3463952.3463962. URL https://www.ifaamas.org/Proceedings/aamas2021/pdfs/p45.pdf.

[172] Uma Roy, Noah Constant, Rami Al-Rfou, Aditya Barua, Aaron Phillips, and Yinfei Yang. Lareqa: Language-agnostic answer retrieval from a multilingual pool. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5919–5930. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.477. URL https://doi.org/10.18653/v1/2020.emnlp-main.477.

[173] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson, 2020. ISBN 9780134610993. URL http://aima.cs.berkeley.edu/.

[174] Magnus Sahlgren. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53, 2008.

[175] Tetsuya Sakai. On the reliability of information retrieval metrics based on graded relevance. *Inf. Process. Manag.*, 43(2):531–548, 2007. doi: 10.1016/j.ipm.2006.07.020. URL https://doi.org/10.1016/j.ipm.2006.07.020.

[176] W.C. Salmon. *Scientific Explanation and the Causal Structure of the World*. Book collections on Project MUSE. Princeton University Press, 1984. ISBN 9780691101705. URL https://books.google.it/books?id=2ug9DwAAQBAJ.

[177] Guillaume Sartoretti, Justin Kerr, Yunfei Shi, Glenn Wagner, T. K. Satish Kumar, Sven Koenig, and Howie Choset. PRIMAL: pathfinding via reinforcement and imitation multi-agent learning. *IEEE Robotics Autom. Lett.*, 4(3):2378–2385, 2019. doi: 10.1109/LRA.2019.2903261. URL https://doi.org/10.1109/LRA.2019.2903261.

[178] Jeff Sauro and James R. Lewis. Correlations among prototypical usability metrics: evidence for the construct of usability. In Dan R. Olsen Jr., Richard B. Arthur, Ken Hinckley, Meredith Ringel Morris, Scott E. Hudson, and Saul Greenberg, editors, *Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI 2009, Boston, MA, USA, April 4-9, 2009*, pages 1609–1618. ACM, 2009. doi: 10.1145/1518701.1518947. URL https://doi.org/10.1145/1518701.1518947.

[179] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Trans. Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605. URL https://doi.org/10.1109/TNN.2008.2005605.

[180] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.05952.

[181] Lenhart Schubert and Matthew Tong. Extracting and evaluating general world knowledge from the brown corpus. In *Proceedings of the HLT-NAACL 2003 Workshop on Text Meaning - Volume 9*, HLT-NAACL-TEXTMEANING '03, page 7–13, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119239.1119241. URL https://doi.org/10.3115/1119239.1119241.

[182] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[183] Wilfrid Sellars. *Science, Perception and Reality*. New York: Humanities Press, 1963.

[184] Machi Shimmei and Noboru Matsuda. Automatic question generation for evidence-based online courseware engineering. In Sergey A. Sosnovsky, Peter Brusilovsky, and Andrew S. Lan, editors, *Proceedings of the Fourth*

International Workshop on Intelligent Textbooks 2022 co-located with 23d International Conference on Artificial Intelligence in Education (AIED 2022), Durham, UK, July 27, 2022, volume 3192 of *CEUR Workshop Proceedings*, pages 18–25. CEUR-WS.org, 2022. URL http://ceur-ws.org/Vol-3192/itb22_p2_short1338.pdf.

[185] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin A. Riedmiller. Deterministic policy gradient algorithms. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 387–395. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/silver14.html.

[186] Amit Singhal, Chris Buckley, and Manclar Mitra. Pivoted document length normalization. *SIGIR Forum*, 51(2):176–184, 2017. doi: 10.1145/3130348.3130365. URL https://doi.org/10.1145/3130348.3130365.

[187] Francesco Sovrano and Fabio Vitali. An objective metric for explainable AI: how and why to estimate the degree of explainability. *CoRR*, abs/2109.05327, 2021. URL https://arxiv.org/abs/2109.05327.

[188] Francesco Sovrano and Fabio Vitali. From philosophy to interfaces: an explanatory method and a tool inspired by achinstein's theory of explanation. In Tracy Hammond, Katrien Verbert, Dennis Parra, Bart P. Knijnenburg, John O'Donovan, and Paul Teale, editors, *IUI '21: 26th International Conference on Intelligent User Interfaces, College Station, TX, USA, April 13-17, 2021*, pages 81–91. ACM, 2021. doi: 10.1145/3397481.3450655. URL https://doi.org/10.1145/3397481.3450655.

[189] Francesco Sovrano and Fabio Vitali. Explanatory artificial intelligence (yai): human-centered explanations of explainable ai and complex data. *Data Mining and Knowledge Discovery*, 2022. doi: 10.1007/s10618-022-00872-x. URL https://doi.org/10.1007/s10618-022-00872-x.

[190] Francesco Sovrano and Fabio Vitali. Generating user-centred explanations via illocutionary question answering: From philosophy to interfaces. *ACM Trans. Interact. Intell. Syst.*, 12(4), nov 2022. ISSN 2160-6455. doi: 10.1145/3519265. URL https://doi.org/10.1145/3519265.

[191] Francesco Sovrano and Fabio Vitali. How to quantify the degree of explainability: Experiments and practical implications. In *31th IEEE International*

Conference on Fuzzy Systems, FUZZ-IEEE 2022, Padova, July 18-23, 2022, pages 1–9. IEEE, 2022.

[192] Francesco Sovrano, Fabio Vitali, and Monica Palmirani. The difference between explainable and explaining: Requirements and challenges under the GDPR. In Grzegorz J. Nalepa, Martin Atzmueller, Michal Araszkiewicz, and Paulo Novais, editors, *Proceedings of the 2nd EXplainable AI in Law Workshop (XAILA 2019) co-located with 32nd International Conference on Legal Knowledge and Information Systems (JURIX 2019), Madrid, Spain, December 11, 2019*, volume 2681 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2681/xaila2019-paper1.pdf.

[193] Francesco Sovrano, Monica Palmirani, and Fabio Vitali. Deep learning based multi-label text classification of UNGA resolutions. In Yannis Charalabidis, Maria Alexandra Cunha, and Demetrios Sarantis, editors, *ICEGOV 2020: 13th International Conference on Theory and Practice of Electronic Governance, Athens, Greece, 23-25 September, 2020*, pages 686–695. ACM, 2020. doi: 10.1145/3428502.3428604. URL https://doi.org/10.1145/3428502.3428604.

[194] Francesco Sovrano, Monica Palmirani, and Fabio Vitali. Legal knowledge extraction for knowledge graph based question-answering. In Serena Villata, Jakub Harasta, and Petr Kremen, editors, *Legal Knowledge and Information Systems - JURIX 2020: The Thirty-third Annual Conference, Brno, Czech Republic, December 9-11, 2020*, volume 334 of *Frontiers in Artificial Intelligence and Applications*, pages 143–153. IOS Press, 2020. doi: 10.3233/FAIA200858. URL https://doi.org/10.3233/FAIA200858.

[195] Francesco Sovrano, Fabio Vitali, and Monica Palmirani. Making things explainable vs explaining: Requirements and challenges under the GDPR. In Víctor Rodríguez-Doncel, Monica Palmirani, Michal Araszkiewicz, Pompeu Casanovas, Ugo Pagallo, and Giovanni Sartor, editors, *AI Approaches to the Complexity of Legal Systems XI-XII - AICOL International Workshops 2018 and 2020: AICOL-XI@JURIX 2018, AICOL-XII@JURIX 2020, XAILA@JURIX 2020, Revised Selected Papers*, volume 13048 of *Lecture Notes in Computer Science*, pages 169–182. Springer, 2020. doi: 10.1007/978-3-030-89811-3_12. URL https://doi.org/10.1007/978-3-030-89811-3_12.

[196] Francesco Sovrano, Fabio Vitali, and Monica Palmirani. Modelling gdpr-compliant explanations for trustworthy AI. In Andrea Ko, Enrico

Francesconi, Gabriele Kotsis, A Min Tjoa, and Ismail Khalil, editors, *Electronic Government and the Information Systems Perspective - 9th International Conference, EGOVIS 2020, Bratislava, Slovakia, September 14-17, 2020, Proceedings*, volume 12394 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2020. doi: 10.1007/978-3-030-58957-8_16. URL https://doi.org/10.1007/978-3-030-58957-8_16.

[197] Francesco Sovrano, Monica Palmirani, Biagio Distefano, Salvatore Sapienza, and Fabio Vitali. A dataset for evaluating legal question answering on private international law. In Juliano Maranhão and Adam Zachary Wyner, editors, *ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021*, pages 230–234. ACM, 2021. URL https://doi.org/10.1145/3462757.3466094.

[198] Francesco Sovrano, Salvatore Sapienza, Monica Palmirani, and Fabio Vitali. A survey on methods and metrics for the assessment of explainability under the proposed AI act. In Schweighofer Erich, editor, *Legal Knowledge and Information Systems - JURIX 2021: The Thirty-fourth Annual Conference, Vilnius, Lithuania, 8-10 December 2021*, volume 346 of *Frontiers in Artificial Intelligence and Applications*, pages 235–242. IOS Press, 2021. doi: 10.3233/FAIA210342. URL https://doi.org/10.3233/FAIA210342.

[199] Francesco Sovrano, Kevin Ashley, Peter Brusilovsky, and Fabio Vitali. Yai4edu: an explanatory AI to generate interactive e-books for education. In Sergey A. Sosnovsky, Peter Brusilovsky, and Andrew S. Lan, editors, *Proceedings of the Fourth International Workshop on Intelligent Textbooks 2022 co-located with 23d International Conference on Artificial Intelligence in Education (AIED 2022), Durham, UK, July 27, 2022*, volume 3192 of *CEUR Workshop Proceedings*, pages 31–39. CEUR-WS.org, 2022. URL http://ceur-ws.org/Vol-3192/itb22_p4_short8391.pdf.

[200] Francesco Sovrano, Monica Palmirani, and Fabio Vitali. Combining shallow and deep learning approaches against data scarcity in legal domains. *Government Information Quarterly*, 39(3):101715, 2022. ISSN 0740-624X. URL https://www.sciencedirect.com/science/article/pii/S0740624X2200048X.

[201] Francesco Sovrano, Alex Raymond, and Amanda Prorok. Explanation-aware experience replay in rule-dense environments. *IEEE Robotics and*

Automation Letters, 7(2):898–905, 2022. doi: 10.1109/LRA.2021.3135927.

[202] Francesco Sovrano, Salvatore Sapienza, Monica Palmirani, and Fabio Vitali. Metrics, explainability and the european ai act proposal. *J*, 5(1): 126–138, 2022. ISSN 2571-8800. doi: 10.3390/j5010010. URL https://www.mdpi.com/2571-8800/5/1/10.

[203] Manfred Stede. Discourse processing. In Lucy Vanderwende, Hal Daumé III, and Katrin Kirchhoff, editors, *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 9-14, 2013, Westin Peachtree Plaza Hotel, Atlanta, Georgia, USA*, pages 4–6. The Association for Computational Linguistics, 2013. URL https://aclanthology.org/N13-4002/.

[204] Nathanael Stein. Causation and explanation in aristotle. *Philosophy Compass*, 6(10):699–707, 2011. doi: https://doi.org/10.1111/j.1747-9991.2011.00436.x. URL https://compass.onlinelibrary.wiley.com/doi/abs/10.1111/j.1747-9991.2011.00436.x.

[205] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng. Learning word representations by jointly modeling syntagmatic and paradigmatic relations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 136–145. The Association for Computer Linguistics, 2015. doi: 10.3115/v1/p15-1014. URL https://doi.org/10.3115/v1/p15-1014.

[206] Peiquan Sun, Wengang Zhou, and Houqiang Li. Attentive experience replay. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 5900–5907. AAAI Press, 2020. URL https://ojs.aaai.org/index.php/AAAI/article/view/6049.

[207] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction*. Adaptive computation and machine learning. MIT Press, 1998. ISBN 978-0-262-19398-6. URL https://www.worldcat.org/oclc/37293240.

[208] Maxwell Szymanski, Martijn Millecamp, and Katrien Verbert. Visual, textual or hybrid: the effect of user expertise on different explanations. In Tracy Hammond, Katrien Verbert, Dennis Parra, Bart P. Knijnenburg, John O'Donovan, and Paul Teale, editors, *IUI '21: 26th International Conference on Intelligent User Interfaces, College Station, TX, USA, April 13-17, 2021*, pages 109–119. ACM, 2021. doi: 10.1145/3397481.3450662. URL https://doi.org/10.1145/3397481.3450662.

[209] Khushboo Thaker, Peter Brusilovsky, and Daqing He. Student modeling with automatic knowledge component extraction for adaptive textbooks. In Sergey A. Sosnovsky, Peter Brusilovsky, Richard G. Baraniuk, Rakesh Agrawal, and Andrew S. Lan, editors, *Proceedings of the First Workshop on Intelligent Textbooks co-located with 20th International Conference on Artificial Intelligence in Education (AIED 2019), Chicago, IL, USA, June 25, 2019*, volume 2384 of *CEUR Workshop Proceedings*, pages 95–102. CEUR-WS.org, 2019. URL http://ceur-ws.org/Vol-2384/paper10.pdf.

[210] Bas C. Van Fraassen. *The Scientific Image*. Clarendon Library of Logic and Philosophy. Clarendon Press, 1980. ISBN 9780198244271. URL https://books.google.it/books?id=VLz2F1zMr9QC.

[211] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. URL doi.org/10.5555/3295222.3295349.

[212] Giulia Vilone and Luca Longo. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Inf. Fusion*, 76:89–106, 2021. doi: 10.1016/j.inffus.2021.05.009. URL https://doi.org/10.1016/j.inffus.2021.05.009.

[213] Giulia Vilone and Luca Longo. A novel human-centred evaluation approach and an argument-based method for explainable artificial intelligence. In Ilias Maglogiannis, Lazaros Iliadis, John Macintyre, and Paulo Cortez, editors, *Artificial Intelligence Applications and Innovations - 18th IFIP WG 12.5 International Conference, AIAI 2022, Hersonissos, Crete, Greece, June 17-20, 2022, Proceedings, Part I*, volume 646 of *IFIP Advances in Information and Communication Technology*, pages 447–460. Springer, 2022. doi: 10.1007/978-3-031-08333-4_36. URL https://doi.org/10.1007/978-3-031-08333-4_36.

[214] Giulia Vilone, Lucas Rizzo, and Luca Longo. A comparative analysis of rule-based, model-agnostic methods for explainable artificial intelligence. In Luca Longo, Lucas Rizzo, Elizabeth Hunter, and Arjun Pakrashi, editors, *Proceedings of The 28th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin, Republic of Ireland, December 7-8, 2020*, volume 2771 of *CEUR Workshop Proceedings*, pages 85–96. CEUR-WS.org, 2020. URL http://ceur-ws.org/Vol-2771/AICS2020_paper_33.pdf.

[215] Andrew Vold and Jack G. Conrad. Using transformers to improve answer retrieval for legal questions. In Juliano Maranhão and Adam Zachary Wyner, editors, *ICAIL '21: Eighteenth International Conference for Artificial Intelligence and Law, São Paulo Brazil, June 21 - 25, 2021*, pages 245–249. ACM, 2021. doi: 10.1145/3462757.3466102. URL https://doi.org/10.1145/3462757.3466102.

[216] Ellen M. Voorhees. The TREC-8 question answering track report. In Ellen M. Voorhees and Donna K. Harman, editors, *Proceedings of The Eighth Text REtrieval Conference, TREC 1999, Gaithersburg, Maryland, USA, November 17-19, 1999*, volume 500-246 of *NIST Special Publication*. National Institute of Standards and Technology (NIST), 1999. URL http://trec.nist.gov/pubs/trec8/papers/qa_report.pdf.

[217] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law and Technology*, 31(2), 2018. doi: 10.2139/ssrn.3063289. URL http://dx.doi.org/10.2139/ssrn.3063289.

[218] Mengdi Wang, Hung Chau, Khushboo Thaker, Peter Brusilovsky, and Daqing He. Knowledge annotation for intelligent textbooks. *Technology, Knowledge and Learning*, 2021. doi: 10.1007/s10758-021-09544-z. URL https://doi.org/10.1007/s10758-021-09544-z.

[219] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2140–2151. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-acl.188. URL https://doi.org/10.18653/v1/2021.findings-acl.188.

[220] Xinru Wang and Ming Yin. Are explanations helpful? A comparative study of the effects of explanations in ai-assisted decision-making. In Tracy Hammond, Katrien Verbert, Dennis Parra, Bart P. Knijnenburg, John O'Donovan, and Paul Teale, editors, *IUI '21: 26th International Conference on Intelligent User Interfaces, College Station, TX, USA, April 13-17, 2021*, pages 318–328. ACM, 2021. doi: 10.1145/3397481.3450650. URL https://doi.org/10.1145/3397481.3450650.

[221] Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3):63:1–63:34, 2021. doi: 10.1145/3386252. URL https://doi.org/10.1145/3386252.

[222] Zichao Wang, Jakob Valdez, Debshila Basu Mallick, and Richard G. Baraniuk. Towards human-like educational question generation with large language models. In Maria Mercedes T. Rodrigo, Noburu Matsuda, Alexandra I. Cristea, and Vania Dimitrova, editors, *Artificial Intelligence in Education - 23rd International Conference, AIED 2022, Durham, UK, July 27-31, 2022, Proceedings, Part I*, volume 13355 of *Lecture Notes in Computer Science*, pages 153–166. Springer, 2022. doi: 10.1007/978-3-031-11644-5_13. URL https://doi.org/10.1007/978-3-031-11644-5_13.

[223] Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. The penn discourse treebank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*, 2019.

[224] Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst continual structured non-stationarity. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11393–11403. PMLR, 2021. URL http://proceedings.mlr.press/v139/xie21c.html.

[225] Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernández Ábrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Multilingual universal sentence encoder for semantic retrieval. In Asli Celikyilmaz and Tsung-Hsien Wen, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, ACL 2020, Online, July 5-10, 2020*, pages 87–94. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-demos.12. URL https://doi.org/10.18653/v1/2020.acl-demos.12.

[226] Haiyan Yin and Sinno Jialin Pan. Knowledge transfer for deep reinforcement learning with hierarchical experience replay. In Satinder Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 1640–1646. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14478.

[227] Daochen Zha, Kwei-Herng Lai, Kaixiong Zhou, and Xia Hu. Experience replay optimization. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4243–4249. ijcai.org, 2019. doi: 10.24963/ijcai.2019/589. URL https://doi.org/10.24963/ijcai.2019/589.

[228] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. *Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms*, pages 321–384. Springer International Publishing, Cham, 2021. ISBN 978-3-030-60990-0. doi: 10.1007/978-3-030-60990-0_12. URL https://doi.org/10.1007/978-3-030-60990-0_12.

[229] Wei Zhu, Wei Zhang, Guo-Zheng Li, Chong He, and Lei Zhang. A study of damp-heat syndrome classification using word2vec and tf-idf. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1415–1420, 2016. doi: 10.1109/BIBM.2016.7822730.

[230] Sandrine Zufferey and Liesbeth Degand. Annotating the meaning of discourse connectives in multilingual corpora. *Corpus Linguistics and Linguistic Theory*, 13(2):399–422, 2017. URL https://doi.org/10.1515/cllt-2013-0022.

[231] Hervé Zwirn and Jean-Paul Delahaye. *Unpredictability and Computational Irreducibility*, pages 273–295. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-35482-3. doi: 10.1007/978-3-642-35482-3_19. URL https://doi.org/10.1007/978-3-642-35482-3_19.

AI Artificial Intelligence

XAI Explainable Artificial Intelligence

YAI Explanatory Artificial Intelligence

DoX Degree of Explainability

GDPR General Data Protection Regulation

AI-HLEG High-Level Expert Group on Artificial Intelligence

SUS System Usability Scale

NCS Need for Cognition Score

TF-IDF Term Frequency–Inverse Document Frequency

EDU Elementary Discourse Unit

AMR Abstract Meaning Representation

NDCG Normalised Discounted Cumulative Gain

MRR Mean Reciprocal Rank

BVA Board of Veterans' Appeals

PTSD Post-Traumatic Stress Disorder

RL Reinforcement Learning

MDP Markov Decision Process

MARL Multi-Agent Reinforcement Learning

GNN Graph Neural Network

XAER Explanation-Aware Experience Replay

DEER Dimensionality-invariant Explanatory Experience Replay

TD Temporal-Difference