Logic-based Machine Learning for Transparent Ethical Agents

Abeer Dyoub[1], Stefania Costantini[1], Francesca A. Lisi[2], and Ivan Letteri[1]⋆

[1] Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
Università degli Studi dell'Aquila, Italy
abeer.dyoub@univaq.it,stefania.costantini@univaq.it,ivan.letteri@univaq.it
[2] Dipartimento di Informatica &
Centro Interdipartimentale di Logica e Applicazioni (CILA)
Università degli Studi di Bari "Aldo Moro", Italy
FrancescaAlessandra.Lisi@uniba.it

Abstract. Autonomous intelligent agents are increasingly engaging in human communities. Thus, they must be expected to follow social and ethical norms of the community in which they are deployed in. In this work we present an approach for developing such ethical agents which are able to develop ethical decision making and judgment capabilities by learning from interactions with the users. Our approach is a logic-based approach and the resulting ethical agents are transparent by design.

1 Introduction

Autonomous intelligent agents are increasingly engaging in human communities. There has been an increasing trend in the last decade to use black box Machine Learning (ML) to develop such agents for critical domains such as healthcare, automotive, and criminal justice, where their decisions deeply impact human lives. Most of the traditional machine learning models are black boxes, as they do not explain their decisions in a way that humans can understand. This lack of transparency and accountability is causing severe harm to society. Examples of such critical applications where ML resulted in a severe consequences are [52], [32] and [50].

The European Union's General Data Protection Regulations (GDPR) [3] is a set of comprehensive regulations for the collection, storage and use of personal information. GDPR took effect as a law across the EU in 2018. This law puts restrictions on automated individual decision-making (that is, algorithms that make decisions based on user-level predictors) which affects users significantly. Furthermore, it is important to mention that the GDPR created a "right to explanation", which allows a user to ask for an explanation of an algorithmic decision that was made about them. This law has posed large challenges for industry, pushing researchers to look for algorithms and evaluation frameworks which avoid discrimination and enable explanation. However, instead of

[3] https://gdpr.eu/

trying to design models that are inherently interpretable by design, most of the recent work concentrates on “Explainable ML”, where a second (posthoc) model is created to explain the first black box model. The posthoc model is an approximation of the original model, thus, explanations are often not reliable because they cannot have perfect fidelity with respect to what original model computes. As the posthoc explanation model is an approximation of the original one, then, given explanations are not always correct [35]. As a result, if we are not sure whether the explanation is correct, we hesitate to trust either the explanation or the original model.

Machine Ethics is an emerging field aiming at creating machines able to compute and choose the best ethical action. Moral judgment and decision making often concern actions that entail some harm, especially loss of life or other physical harm, loss of rightful property, loss of privacy, or other threats to autonomy. Moral judgments are also triggered by actions that affect not only the actor but others as well. When autonomous agents are to be deployed in sensitive environments like, e.g., healthcare, their behavior should be ethically constrained. In other words, those agents must be designed on ethical bases. Moral decision making and judgment is a complicated process involving many aspects: it is considered as a mixture of reasoning and emotions. In addition, moral decision making is highly flexible, contextual and culturally diverse. Since the beginning of this century there have been several attempts for implementing ethical decision making into intelligent autonomous agents, using different approaches. So far however, no fully descriptive and widely acceptable model of moral judgment and decision making exists. None of the developed solutions seem to be fully convincing to provide a trusted moral behavior. Approaches to machine ethics are classified into *top-down* approaches, which try to implement specific normative theory of ethics into the autonomous agent so as to ensure that the agent acts in accordance with the principles of this theory. The *bottom-up* approaches are developmental or learning approaches, in which ethical mental models emerge via the activity of individuals rather than expressed explicitly in terms of normative theories of ethics [51]. In other words, generalism versus particularism, principles versus case based reasoning. Both approaches to morality have advantages and disadvantages. We need hybrid approaches that combine both points of view in one framework.

Transparency is a key requirement for ethical machines, because eventually the relevant criteria for an AI system to be considered ethical will involve the system to be able to explain and justify its behavior to users or to the society as a whole. From transparency flow two attributes of particular importance in the machine ethics field and AI field in general, viz. trust and accountability. It is hard to trust a machine unless you have some understanding of what it is doing and why. Furthermore, without transparency it becomes very difficult to understand who is responsible when a machine does not behave as we expect it to. Interpretability empowers safety by enabling ML systems models to be tested, audited, and debugged which leads to increased safety of such systems [34]. Interpretability helps to detect unethical problems like bias that ML models learned, either from wrong data pre-processing or due to wrong settings parametrization, which arises from incompleteness of the problem definition. We believe however that the best way to design transparent ethical agents is the use of ML models that are inherently-interpretable (transparent by design), they provide their own explanations, which are faithful to what the model actually computes, instead of trying to explain ML black-box models.

In this work, we present a logic-based hybrid approach for implementing transparent ethical agents. The proposed approach combines deductive (rule-based) logic programming and inductive (learning) logic programming approaches in one framework for building our ethical agent. We use Answer Set Programming (ASP) for knowledge representation and reasoning, and Inductive Logic Programming (ILP) as a machine learning technique for learning from cases and generating the missing detailed ethical rules needed for reasoning about future similar cases. The newly learned rules are to be added to the agent knowledge base. ASP, a purely declarative non-monotonic reasoning paradigm, was chosen because ethical rules are known to be default rules, which means that they tolerate exceptions. This in fact nominates non-monotonic logics which simulate common sense reasoning to be used for formalizing different ethical conceptions. In addition, there are the many advantages of ASP including it is expressiveness, flexibility, extensibility, ease of maintenance, readability of its code. The availability of free *solvers* to derive consequences of different ethical principles automatically can help in precise comparison of ethical theories, and makes it easy to validate our models in different situations. ILP was chosen as a machine learning approach because it supports two very important and desired aspects of machine ethics implementation into artificial agents viz. explainability and accountability, ILP is known for its explanatory power, elements of the generated rules can be used to formulate an explanation for the choice of certain decisions over others. Comprehensibility of logic-based representations is in fact one of their most recognized advantages. Thus, the resulting agents are transparent by design. Moreover, ILP also seems better suited than statistical methods to domains in which training examples are scarce as in the case of ethical domain.

After providing some background on the adopted techniques and a discussion of related work, we present our approach considering as a sample application domain online customer service (so-called "chatbots"). However, the approach is general enough to be adopted to implement ethical agents in different domains.

2 Background

2.1 Answer Set Programming (ASP) in a Nutshell

ASP is a logic programming paradigm under answer set (or "stable model") semantics [24], which applies ideas of autoepistemic logic and default logic. In ASP, search problems are reduced to computing answer sets, and an *answer set solver* (i.e., a program for generating stable models) is used to find solutions. An answer set Program is a collection of rules of the form: $H \leftarrow A_1, \dots, A_m, notA_{m+1}, \dots, notA_n$ were each of A_i's is a literal in the sense of classical logic. Intuitively the above rule means that if $A_1, \dots, A_m$ are true and if $A_{m+1}, \dots, A_n$ can be safely assumed to be false then H must be true. The left-hand side and right-hand side of rules are called *head* and *body*, respectively. A rule with empty body ($n = 0$) is called a *fact*. A rule with empty head is a *constraint*, and states that literals of the body cannot be simultaneously true in any answer set. Unlike other semantics, a program may have several answer sets or may have no answer set. So, differently from traditional logic programming, the solutions of a problem are not obtained through substitutions of variables values in answer to a query. Rather, a program Π describes a problem, of which its answer sets represent the possible solutions, found by means of ASP *solvers* [4]. For more information about ASP and its applications (also to the agents realm) the reader can refer, among many, to [10, 18] and the references therein.

2.2 Inductive Logic Programming in a Nutshell

ILP [36] is a branch of artificial intelligence (AI) which investigates the inductive construction of logical theories from examples and background knowledge. In the general settings, we assume a set of Examples *E*, positive E^+ and negative E^-, and some background knowledge *B*. An ILP algorithm finds the hypothesis *H* such that $B \bigcup H \models E^+$ and $B \bigcup H \not\models E^-$. The possible hypothesis space is often restricted with a language bias that is specified by a series of mode declarations *M*. A mode declaration is either a head declaration *modeh(r, s)* or a body declaration *modeb(r, s)*, where *s* is a ground literal, this scheme serves as a template for literals in the head or body of a hypothesis clause, where *r* is an integer, the recall, which limits how often the scheme can be used. A scheme can contain special *placemarker* terms of the form $\sharp$ *type*, *+type* and *-type*, which stand, respectively, for ground terms, input terms and output terms of a predicate *type*. Finally, it is important to mention that ILP has found applications in many areas. For more information on ILP and applications, refer, among many to [37] and references therein.

ILP has received a growing interest over the last two decades. ILP has many advantages over statistical machine learning approaches: the learned hypotheses can be easily expressed in plain English and explained to a human user, and it is possible to reason with the learned knowledge. Most of the work on ILP frameworks has focused on learning definite logic programs (e.g. [49], [42]) and normal logic programs (e.g. [15]). In the last decade, several new learning frameworks and algorithms have been introduced for learning under the answer set semantics. ASPAL [16] is the first ILP system to learn answer set programs, by encoding ILP problems as ASP programs, and having an ASP solver find the hypothesis. Then followed by many others, see e.g. [28], [47], [46], [27].

3 Related Work: Logic for Programming Machine Ethics

Logic-based approaches have a great potential to model moral machines, in particular via non-monotonic logics. Ethical theories and dilemmas have always been represented in a declarative form by ethicists, who also used formal and in-formal logic to reason about them. Logical representations help to make ideas clear and highlight differences between different ethical systems.

Tom Powers in [41] assesses the viability of using deontic and default logics, to implement Kant's categorical imperative. Kant's categorical imperative ('Act only according to that maxim, whereby you can at the same time, will that it should become

[4] Many performant ASP solvers an Prolog interpreters are freely available, a list of them is reported at `https://en.wikipedia.org/wiki/Answer_set_programming`

a universal law without contradiction' [39]). Three views on how to computationally model categorical imperative are envisaged: First, in order for a machine to maintain consistency in testing ethical behavior, construct a moral theory for individual maxims, and map them onto deontic categories. Deontic logic is regarded as an appropriate formalism with respect to this first view. Second, there is the need for commonsense reasoning in the categorical imperative, to deal with contradiction. For this view, he refers to non-monotonic logic, which is appropriate to capture defeating conditions to a maxim. Default logic of Reiter [43] is regarded as a suitable formalism. Third, the construction of a coherent system of maxims, where the author sees such construction analogous to the belief revision problems. In the context of bottom-up construction, he envisages an update procedure for a machine to update its system of maxims with another maxim, though it is unclear to him how such an update can be accomplished. His formalisms in these three views were only considered abstractly, and no implementation is referred to address them.

In [11], the authors suggest that mechanized multi-agent deontic logics might be an appropriate vehicle for engineering ethically correct robot behaviors. They use the logical framework Athena [5], to encode a natural deduction system of Murakami [38] axiomatization of Horty's utilitarian formulation of multi-agent deontic logic [26]. The use of an interactive theorem prover is motivated by the idea that an agents operate according to ethical codes bestowed on them, and when its automated reasoning fails, it suspends its operation and asks human guidance to resolve the issue. Taking an example in health care, where two agents are in charge of two patients with different needs (patient H1 depends on life support, whereas patient H2 on very costly pain medication), two actions are considered: (1) terminate H1's life support to secure his organ for five humans; and (2) delay delivery of medication to H2 to conserve hospital resources. It starts by supposing several candidates of ethical codes, from harsh utilitarian (that both terminates H1's life and delay H2 medication) to most benevolent (neither terminates H1's life nor delay H2 medication); these ethical codes are formalized using the aforementioned deontic logics. The logic additionally formalizes behaviors of agents and their respective moral outcomes. Given these formalizations, Athena is employed to query each ethical code candidate in order to decide which amongst them should be operative, meaning that the best moral outcome (viz., that resulting from neither terminating H1's life nor delaying H2 medication) is provable from the operative one.

Other attempts tried to formalize ethical systems using modal logic formalisms [25] and then trying to operationalize these formalizations on computer, like in [11] and [41]. These formalizations are mainly based on the use of deontic logics [33], that are well adapted to ethical systems focused on laws were permission and prohibitions are well defined, but not to consequentialist ethical systems.

Pereira and Saptawijaya have proposed the use of different logic-based features, for representing diverse issues of moral facets, such as moral permissibility, doctrines of Double Effect and Triple Effect, the Dual-process Model, counterfactual thinking in moral reasoning. They investigated the use of abduction, probabilistic logic programming, logic programming updating, tabling. These logic-based reasoning features were synthesized in three different systems: ACORDA, Probabilistic EPA, QUALM [40].

One way for implementing ethical decision making is qualifying and quantifying the good and the bad ramifications of ethical decisions before taking them. This task is non-trivial, as there could be many approaches for doing this. First qualifying the 'Good' involves identifying modes for defining the 'Good' which is a controversial task because there exist a lot of theories attempting to define the 'Good'. [8] presents a model for quantifying the good after it has been qualified. For Qualifying the good, they present two modes, one based on rights and the other one is based on values. For quantifying the good, they propose a method in which they define three weighing parameters for the good and the bad ramifications of events caused by actions. Then, they integrate all weights into a single number, which represents the weight of an event in relation to a particular modality and group of people. The total weight of an event then is the difference between the sums of all its weighted good and bad ramifications. Greater weights correspond to more participation in the Good, while negative weights do more harm than good. Their approach was implemented in ASP.

In [23], the authors formalized three ethical conceptions (the Aristotelian rules, Kantian categorical imperative, and Constant's objection) using nonmonotonic logic, particularly Answer Set Programming.

In [13], authors introduced a model that can be used by the agent in order to judge the ethical dimensions of its own behavior and the behavior of others. Their model was implemented in ASP. However, the model is still based on a qualitative approach. Whereas it can define several moral valuations, there is neither a degree of desires, nor a degree of capability, nor a degree of rightfulness. Moreover, ethical principles need to be more precisely defined to capture various sets of theories suggested by philosophers.

Sergot in [45], provides an alternative representation to the argumentative representation of a moral dilemma case concerning a group of diabetic persons, presented in [6], where the authors used value-based argumentation to solve this dilemma. According to Sergot, the argumentation framework representation doesn't work well and doesn't scale. Sergot proposal for handling this kind of dilemmas is based on Defeasible Conditional Imperatives. The proposed solution was implemented in ASP.

JEREMY [3] is an implementation of the Hedonistic Act Utilitarianism. This theory states that an action is morally right if and only if that action maximizes the pleasure, i.e. the one with the greatest net pleasure consequences, taking into account those affected by the action. The theory of Act Utilitarianism has, however, been questioned as not entirely agreeing with intuition. The authors of JEREMY, to respond to critics of act utilitarianism, have created another system, W.D. [3] which avoids a single absolute duty, by following several duties. Their system follows the theory of prima facie duties of Ross [44] and is implemented in ILP. Ethics is more complicated than following a single ethical principle. According to Ross ([44]), ethical decision making involves considering several Prima Facie duties, and any single-principled ethical theory like Act Utilitarianism is sentenced to fail. ILP was used by researchers to model ethical decision making in MedEthEx [4], and EthEl [1]. These two systems are based on a more specific theory of prima facie duties viz., the principle of Biomedical ethics of Beauchamp and Childress [7]. In these systems, the strength of each duty is measured by assigning it a weight, capturing the view that a duty may take precedence over another. Then computes, for each possible action, the weighted sum of duty satisfaction, and the right action is the one with the greatest sum. The three systems use ILP to learn the relation *supersedes(A1,A2)* which says that action *A1* is preferred over action *A2* in an ethical dilemma involving these choices. MedEthEx is designed to give advice for dilemmas in biomedical fields, while EthEl is applied to the domain of eldercare with the main purpose to remind a patient to take her medication, taking ethical duties into consideration. GenEth [2] is another System that makes use of ILP. GenEth has been used to codify principles in a number of domains relevant to the behavior of autonomous systems.

4 Proposed Approach for Ethical Agent Development

Embedding norms in autonomous intelligent systems requires a clear outlining of the community in which they are to be deployed. In fact, determining which moral values to aim for and which ethical principles to adhere to in a given circumstances is one of the main challenges for ethical reasoning. Codes of ethics and conduct provide us with a framework to work within. However, enforcing codes of conduct and ethics in our intelligent agents is not an easy task. These codes are mostly abstract and based upon general principles such as confidentiality, accountability, honesty, inclusiveness, empathy, fidelity, etc., that are quite difficult to put into practice. Moreover, abstract principles such as these may contain terms whose meaning may change according to the context. It is difficult to use deductive logic only to address such a problem: it is in fact hardly possible for experts to define fine-grained detailed rules to cover all possible situations.

We need to teach our machines the codes of ethics and conduct of the domain in which they need to be deployed. Artificial agents in fact could, similarly to humans, acquire ethical decision making and judgment capabilities by implicit processes, in particular via inductive learning [51]. With increasing autonomy, there will be more situations that require morally relevant decisions to be made by the artificial agent. Many of these decisions cannot be foreseen in details. Therefore, we need bottom-up (learning) approaches because it is difficult to fully specify all possible scenarios in advance (framing problem), and because no actual agreement about explicit theory of normative ethics to be implemented [14].

In this section we present our approach. The application that we have considered as a case study is online customer service ("chatbots"). In this work we are concerned only with the ethical reasoning capabilities of our agent, other details related to the complete design of a chatbot are not handled here, for more details refer [22]. The behavior of an ethical online customer service chatbot should be dictated by the codes of ethics and conduct of her company. Codes of ethics in domains such as customer service are abstract general principles, therefore they apply to a wide range of situations. They are subject to interpretations and may have different meanings in different contexts. There are no intermediate rules that elaborate these abstract principles or explain how they apply in concrete situations. We propose an approach to generate these intermediate rules from interactions with clients through a simplified dialogue.

Initially our agent will have in her knowledge base the domain knowledge, together with a small ethical background knowledge limited to few ethical general rules represented by ASP like:

$$rule1 = \{unethical(V) \leftarrow not\_correct(V), answer(V).\}$$

which says that it is unethical to provide incorrect information to the customers. The missing ethical rules are learned by our agent incrementally overtime through interactions with clients. The newly generated rules are to be added to our agent knowledge base, to be used for ethical reasoning of future cases.

4.1 Formalizing Ethical Rules via ASP

Ethical principles are rules of behavior. In other words, rules that help us to decide what is an ethical action, and what is not ethical. In addition, they help us to ethically judge and evaluate the behavior of others. Thus, any ethical system, i.e., any consistent set of ethical principles, needs defining a decision making procedure.

Considering the domain of interest (online customer service), we want to describe these decision making procedures in a purely declarative way. In fact, by using the ASP formalism, it is possible to model ethical rules explaining the status of a certain case situation (or a set of similar cases). To show an example of the ASP-based formalism adopted in this work, let us consider the following scenarios were we want to teach our customer service chatbot that any claim it does should be backed by genuine scientific evidence. For example, marketing certain products as healthy way to loose weight, or healthy way to remove hair, etc, while there is not significant evidence to support such claim, is considered unethical practice.

Example1: Q1: I need a product to loose weight. A1: I suggest you productX. We claim that productX is a healthy way to loose weight. ProductX costs 10Euros. evaluation: unethical answer.

Example2: Q2: I need a product to loose weight. A2: I suggest you productX. We claim that productX is a healthy way to loose weight. We have a verified scientific certificate for our claim. ProductX costs 10Euros. evaluation: ethical answer.
In the example1 the answer is un ethical because the claim is not supported by a verified scientific certificate.

The set of facts of the sample case scenario are:
productX costs 10Euros. claim productX is a healthy way to loose weight
Their corresponding ASP translations are:
cost(productX,10). claim(productXisHealthyWayToLooseWeight)

It is useful to start the ethical analysis of the case with the question: what are the relevant facts to be considered in the ethical evaluation of the answer?. At least one fact of every scenario's set of facts is the questioned fact, i.e. the fact corresponding to the ethical question raised in the scenario. So, in this example, the third fact namely, 'claim productX is a healthy way to loose weight' is the questioned fact [5]. This because, as mentioned earlier, it is unethical to claim something about a product without having a significant scientific certificate to support such claim. So now, using ASP formalism, this can be expressed with the following rule:

[5] Case scenarios are analyzed by competent ethical judges of the domain and an ethical evaluation is provided for each scenario

$$unethical(A) \leftarrow answer(A), claim(A), not\ verifiedcertificate(A).$$

The above rule represents the knowledge that the predicate $unethical(A)$, denoting that an answer A is unethical if it includes the use of a claim, and this use is not supported by a scientific certificate in this case. An answer set program Π containing the above rule, along with the fact
$answer(productXisHealthyWayToLooseWeight)$, and the fact
$claim(productXisHealthyWayToLooseWeight)$, and nothing says that this claim is supported by verified certificate, which can be safely assumed false in case there is no information about it, will logically entail ($\models$) that this answer is unethical. However, if we add the following fact in the program Π:
$verifiedcertificate(productXisHealthyWayToLooseWeight)$, then the program Π will no longer entails $unethical(productXisHealthyWayToLooseWeight)$. Finally to complete the evaluation program we add the rules:

$$ethical(A) \leftarrow answer(A), not\ unethical(A). \leftarrow ethical(A), unethical(A).$$

Which says that an answer is ethical if it is not known to be unethical (i.e no knowledge about the contrary), and an answer cannot be ethical and unethical at the same time.

Assume to add the following definition of the $verifiedcertificate(A)$ predicate:

$$-verifiedcertificate(A) \leftarrow answer(A), not\ verifiedcertificate(A).$$
$$\leftarrow verifiedcertificate(A), -verifiedcertificate(A).$$

which says that an answer is not supported with a verified certificate if it is not known to be supported with a verified certificate. The program Π will have the following answer set (model):

$M_1 = \{claim(productXisHealthyWayToLooseWeight),$
$answer(productXisHealthyWayToLooseWeight),$
$-verifiedcertificate(productXisHealthyWayToLooseWeight),$
$unethical(productXisHealthyWayToLooseWeight)\}$

4.2 Learning ASP Ethical Rules from Interactions with Users

During the training phase, the trainer enters a series of sentences in the form of requests and responses through the keyboard simulating a customer service chat point conversation, along with the ethical evaluation of the responses in each scenario. The first step is to convert the natural language sentences into the syntax of ASP. The system remembers the facts about the narratives given by the trainer and learns to form ethical evaluation rules according to the facts given in the story context (*C*) and background knowledge (*B*). For learning the ethical rules (*H*) needed for dictating the ethical behavior of our agent, we use the state of the art ILP tool ILED [27] . In the test phase, the agent uses both *B & H* to respond to the client request avoiding unethical practices. The goal is to recognize unethical responses from combinations of case' facts.

To do so, the trainer will provide the system with different positive and negative examples; table 2 demonstrate the learning process. The system will start constructing hypotheses from the first available case (c1). A generated hypothesis (rule) will be added to the agent knowledge base. When a new case (c2) arrives, the system will check whether the new case is covered by the running hypothesis. If not, it will start the revision process to update the running hypothesis (rule) to a new rule that cover the new case (see table 2). Table 1 shows the background knowledge and the mode declarations serving as patterns for restricting the hypotheses search space. For more details about our approach the reader may refer to [21, 20].

Input(B,M)	
Mode Declaration M	**Background Knowledge** B
modeh(unethical(+answer)).	$notclaim(X) \leftarrow$
modeb(claim(+answer)).	*not claim(X),answer(X).*
modeb(notclaim(+answer)).	$notverifiedcertificate(X) \leftarrow$
modeb(verifiedcertificate(+answer)).	*not verifiedcertificate(X),claim(X).*
modeb(notverifiedcertificate(+answer)).	

Table 1. Example: ILED input (B and M)

5 Discussion and Conclusion

In the context of many application domains and a fortiori in domains involving ethical aspects, it is crucial that systems' decisions are transparent and comprehensible and in consequence trustworthy. Comprehensibility is one of the main features that distinguish logic-based representations from those proper of statistical ML. Logic programs are comprehensible by humans, and they have a well-defined declarative and operational semantics.

Providing explanations to system's decisions is fundamentally linked to its reliability and trustworthiness. A sound explanation guarantees that the correlations extracted by the algorithm from the data are causal relations that have sense in the considered system. Logic Programming is able to model causality which is crucial especially for ethical reasoning.

The ethical agent mentioned in section 4 consists of a set of modules [22, 19]. The ethical reasoning module is nothing but an ASP module, consisting of a set of ASP rules and facts describing the ontology of the domain (facts and initial general ethical rules of the domain encoded deductively using ASP), in addition to the newly learned rules. When this agent is affronted with a new case scenario, the case facts are extracted and added to the ASP-reasoner knowledge base. Then, an ASP-solver will output a model (answer set). This model includes the ethical evaluation result as well as the cause of this result, in other words, the justification for the conclusion computed by the ethical agent. Going back to subsection 4.1, the output (the model) given by the solver says that the answer of the chat agent is 'unethical' in the illustrated case scenario. The output model contains other facts, these facts are the cause for this evaluation. This result is shown to the user [22].

The interface which shows the evaluation results to the user has been recently extended to explicitly show the justification/explanation behind this evaluation extracted from the answer set, as for example:
Result: unethical answer healthy way to loose weight productX.

Justification: because you claim healthy way to loose weight productX, and no verified certificate to healthy way to loose weight productX.
In fact, the ASP-program models contain both the output and the justification for the given output, which can be easily shown to the user. No further processing is in required to generate the explanations for the users, as such explanations are already part of the output model.

ILP, as a logic-based ML paradigm which induces logic programs from data, has shown a great potential for addressing limitations of standard ML approaches concerning opaqueness, poor generalization, and need for a huge quantity of training data. ILP complements deductive programming approaches [9, 30]. In cases where it is hard for human inductive reasoning to syntheses a specific algorithm details, ILP can be used to induce program candidates from user-provided data or test cases [48]. ILP does not require huge amounts of training examples such as other (statistical) Machine Learning methods and produces interpretable results, that means a set of rules which can be analyzed and adjusted if necessary. So, ILP appears to be a suitable and promising technique for implementing machine ethics, where scarcity of examples is one of the main challenges, and comprehensibility of the output is indispensable.

Combining logic-based representation and logic-based learning for modeling ethical agents, as done in our aforementioned work, provides many advantages: increases the reasoning capability of agents; promotes the adoption of hybrid strategies that allow both top-down design and bottom-up learning via context-sensitive adaptation of models of ethical behavior; allows the generation of rules with valuable expressive and explanatory power, thus equipping agents with the capacity to make ethical decisions, and to explain the reasons behind these decisions.

In our opinion and for the sake of transparency, ethical decision-making and judgment should however be guided by explicit ethical rules determined by competent judges or ethicists, or generated automatically but approved through consensus of ethicists. Machine learning models should follow an explainability-by-design approach able to provide explanations to users, together with adoption of and regulatory bodies as a requirement from the very beginning of the life cycle of the product. This becomes critical especially when personal data are involved or when the system can cause harm or violations to fundamental rights of users.

In conclusion, we believe that logic-based approaches, which are inherently-interpretable, have a great potential for implementing ethical machines, avoiding the potential problems caused by black-box ML models. This especially in consideration of the recent advances in Inductive Logic Programming [17] which puts it in a position to substitute black-box machine learning, particularly in critical applications. An interesting immediate future direction for our work is in fact the exploitation of the results of [29] which proposes a new tool, ILASP, for learning ASP program fragments.

References

1. Anderson, M., Anderson, S.L.: ETHEL: toward a principled ethical eldercare system. In: AI in Eldercare: New Solutions to Old Problems, Papers from the 2008 AAAI Fall Symposium, Arlington, Virginia, USA, November 7-9, 2008. AAAI Technical Report, vol. FS-08-

window w1	
Facts *answer(healthyWayToRemoveHairproduct3).* *claim(healthyWayToRemoveHairproduct3).* *notverifiedcertificate(healthyWayToRemoveHairproduct3).*	**Conclusion** *unethical(healthyWayToRemoveHairproduct3).*
Kernal Set unethical(healthyWayToRemoveHairproduct3) ← answer(healthyWayToRemoveHairproduct3), claim(healthyWayToRemoveHairproduct3). notverifiedcertificate(healthyWayToRemoveHairproduct3).	**Variabilized Kernal Set** K1= unethical(V) ← answer(V), claim(V). notverifiedcertificate(V).
Running Hypothesis $H1$= unethical(V) ← answer(V).	**Support Set** $H1.supp = \{K1\}$
window w2	

Facts
answer(healthyWayToLooseWeightproduct1).
claim(healthyWayToLooseWeightproduct1)
notverifiedcertificate(healthyWayToLooseWeightproduct1)
	Conclusion
	unethical(healthyWayToLooseWeightproduct1).

Kernal Set unethical(healthyWayToLooseWeightproduct1) ← answer(healthyWayToLooseWeightproduct1), claim(healthyWayToLooseWeightproduct1), notverifiedcertificate(healthyWayToLooseWeightproduct1).	**Variabilized Kernal Set** K2= unethical(X1) ← answer(X1), claim(X1), notverifiedcertificate(X1).
Running Hypothesis: Remains unchanged	**Support Set:** $H2.supp = \{K1, K2\}$
window w3	

Facts
answer(healthyWayToLooseWeightproduct2).
notclaim(healthyWayToLooseWeightproduct2)
verifiedcertificate(healthyWayToLooseWeightproduct2)
	Conclusion
	notunethical(healthyWayToLooseWeightproduct2).

Revised Hypothesis $H3$= unethical(X1) ←answer(X1), notverifiedcertificate(X1).	**Support Set** $H3.supp = \{K1, K2\}$
window w4	

Facts
answer(healthyWayToLooseWeightproduct4).
claim(healthyWayToLooseWeightproduct4)
verifiedcertificate(healthyWayToLooseWeightproduct4)
	Conclusion
	notunethical(healthyWayToLooseWeightproduct4).

Running Hypothesis: Remains unchanged	**Support Set:** $H4.supp = \{K1, K2\}$
window w5	
Facts *answer(healthyWayToRemoveHairproduct5).* *notclaim(healthyWayToRemoveHairproduct5).* *notverifiedcertificate(healthyWayToRemoveHairproduct5).*	**Conclusion** *notunethical(healthyWayToRemoveHairproduct5).*
Revised Hypothesis $H5$= unethical(X1) ←answer(X1), notverifiedcertificate(X1),claim(X1).	**Support Set** $H5.supp = \{K1, K2\}$

Table 2. Example:Input examples and output theory

02, pp. 4–11. AAAI (2008), http://www.aaai.org/Library/Symposia/Fall/fs08-02.php

2. Anderson, M., Anderson, S.L.: Geneth: A general ethical dilemma analyzer. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada. pp. 253–261. AAAI Press (2014). https://doi.org/10.1515/pjbr-2018-0024
3. Anderson, M., Anderson, S.L., Armen, C.: Towards machine ethics. In: AAAI-04 workshop on agent organizations: theory and practice, San Jose, CA (2004)
4. Anderson, M., Anderson, S.L., Armen, C.: Medethex: Toward a medical ethics advisor. In: Caring Machines: AI in Eldercare, Papers from the 2005 AAAI Fall Symposium, Arlington, Virginia, USA, November 4-6, 2005. AAAI Technical Report, vol. FS-05-02, pp. 9–16. AAAI Press (2005), https://www.aaai.org/Library/Symposia/Fall/fs05-02.php
5. Arkoudas, K., Bringsjord, S., Bello, P.: Toward ethical robots via mechanized deontic logic. In: AAAI Fall Symposium on Machine Ethics. pp. 17–23 (2005)
6. Atkinson, K., Bench-Capon, T.J.M.: Addressing moral problems through practical reasoning. In: Deontic Logic and Artificial Normative Systems, 8th International Workshop on Deontic Logic in Computer Science, DEON 2006, Utrecht, The Netherlands, July 12-14, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4048, pp. 8–23. Springer (2006). https://doi.org/10.1007/11786849_4
7. Beauchamp, T.L., Childless, J.F.: Principles of biomedical ethics. International Clinical Psychopharmacology **6**(2), 129–130 (1991). https://doi.org/10.1001/jama.1984.03340360075041
8. Berreby, F., Bourgne, G., Ganascia, J.: A declarative modular framework for representing and applying ethical principles. In: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017. pp. 96–104. ACM (2017), http://dl.acm.org/citation.cfm?id=3091145
9. Bodík, R., Torlak, E.: Synthesizing programs with constraint solvers. In: Computer Aided Verification - 24th International Conference, CAV 2012, Berkeley, CA, USA, July 7-13, 2012 Proceedings. Lecture Notes in Computer Science, vol. 7358, p. 3. Springer (2012). https://doi.org/10.1007/978-3-642-31424-7_3
10. Gerhard Brewka, Thomas Eiter and Miroslaw Truszczynski (eds.) Answer Set Programming: Special Issue. AI Magazine, **3**(3) (2016)
11. Bringsjord, S., Arkoudas, K., Bello, P.: Toward a general logicist methodology for engineering ethically correct robots. IEEE Intelligent Systems **21**(4), 38–44 (2006), https://doi.org/10.1109/MIS.2006.82
12. Chollet, F.: On the measure of intelligence. CoRR **abs/1911.01547** (2019), http://arxiv.org/abs/1911.01547
13. Cointe, N., Bonnet, G., Boissier, O.: Ethical judgment of agents' behaviors in multi-agent systems. In: Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016. pp. 1106–1114. ACM (2016), http://dl.acm.org/citation.cfm?id=2937086
14. Conitzer, V., Sinnott-Armstrong, W., Borg, J.S., Deng, Y., Kramer, M.: Moral decision making frameworks for artificial intelligence. In: Singh, S.P., Markovitch, S. (eds.) Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA. pp. 4831–4835. AAAI Press (2017), http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14651
15. Corapi, D., Russo, A., Lupu, E.: Inductive logic programming as abductive search. In: Technical Communications of the 26th International Conference on Logic Programming, ICLP 2010, July 16-19, 2010, Edinburgh, Scotland, UK. LIPIcs, vol. 7, pp. 54–63. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2010)

16. Corapi, D., Russo, A., Lupu, E.: Inductive logic programming in answer set programming. In: Inductive Logic Programming - 21st International Conference, ILP 2011, Windsor Great Park, UK, July 31 - August 3, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7207, pp. 91–97. Springer (2012)
17. Cropper, A., Dumancic, S., Muggleton, S.H.: Turning 30: New ideas in inductive logic programming. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020. pp. 4833–4839. ijcai.org (2020). https://doi.org/10.24963/ijcai.2020/673
18. Dyoub, A., Costantini, S., Gasperis, G.D.: Answer set programming and agents. Knowledge Eng. Review **33**, e19 (2018). https://doi.org/10.1017/S0269888918000164
19. Dyoub, A., Costantini, S., Lisi, F.A.: An approach towards ethical chatbots in customer service. In: Proceedings of the 6th Italian Workshop on Artificial Intelligence and Robotics co-located with the XVIII International Conference of the Italian Association for Artificial Intelligence (AI*IA 2019), Rende, Italy, November 22, 2019. CEUR Workshop Proceedings, vol. 2594, pp. 1–5. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2594`
20. Dyoub, A., Costantini, S., Lisi, F.A.: Learning Answer Set Programming Rules for Ethical Machines. In: Proceedings of the Thirty Fourth Italian Conference on Computational Logic-CILC, June 19-21, 2019, Trieste, Italy. CEUR-WS.org (2019), `http://ceur-ws.org/Vol-2396/`
21. Dyoub, A., Costantini, S., Lisi, F.A.: Towards an ILP application in machine ethics. In: Inductive Logic Programming - 29th International Conference, ILP 2019, Plovdiv, Bulgaria, September 3-5, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11770, pp. 26–35. Springer (2019). https://doi.org/10.1007/978-3-030-49210-6
22. Dyoub, A., Costantini, S., Lisi, F.A., Gasperis, G.D.: Demo paper: Monitoring and evaluation of ethical behavior in dialog systems. In: Advances in Practical Applications of Agents, Multi-Agent Systems, and Trustworthiness. The PAAMS Collection - 18th International Conference, PAAMS 2020, L'Aquila, Italy, October 7-9, 2020, Proceedings. Lecture Notes in Computer Science, vol. 12092, pp. 403–407. Springer (2020). https://doi.org/10.1007/978-3-030-49778-1_35
23. Ganascia, J.G.: Modelling ethical rules of lying with answer set programming. Ethics and information technology **9**(1), 39–47 (2007). https://doi.org/10.1007/s10676-006-9134-y
24. Gelfond, M., Lifschitz, V.: The stable model semantics for logic programming. In: Kowalski, R., Bowen, K. (eds.) Proc. of the 5th Intl. Conf. and Symposium on Logic Programming. pp. 1070–1080. MIT Press (1988)
25. Gensler, H.J.: Formal Ethics. Psychology Press (1996)
26. Horty, J.F.: Agency and deontic logic. Oxford University Press (2001)
27. Katzouris, N., Artikis, A., Paliouras, G.: Incremental learning of event definitions with inductive logic programming. Machine Learning **100**(2-3), 555–585 (2015). https://doi.org/10.1007/s10994-015-5512-1
28. Law, M., Russo, A., Broda, K.: Iterative learning of answer set programs from context dependent examples. TPLP **16**(5-6), 834–848 (2016)
29. Law, M., Russo, A., Broda, K.: The ILASP system for inductive learning of answer set programs. CoRR **abs/2005.00904** (2020), `https://arxiv.org/abs/2005.00904`
30. Manna, Z., Waldinger, R.J.: A deductive approach to program synthesis. ACM Trans. Program. Lang. Syst. **2**(1), 90–121 (1980). https://doi.org/10.1145/357084.357090
31. Marcus, G.: Deep learning: A critical appraisal. CoRR **abs/1801.00631** (2018), `http://arxiv.org/abs/1801.00631`
32. MCGOUGH, M.: How bad is Sacramento's air, exactly? Google results appear at odds with reality, some say. Sacramento Bee (2018), `https://www.sacbee.com/news/california/fires/article216227775.html`

33. Meyer, J.J.C., Dignum, F., Wieringa, R.J.: The paradoxes of deontic logic revisited: a computer science perspective. Technical Report (UU-CS-1994-38) (1994)
34. Miller, T.: Explanation in artificial intelligence: Insights from the social sciences. Artificial Intelligence **267**, 1–38 (2019)
35. Mittelstadt, B.D., Russell, C., Wachter, S.: Explaining explanations in AI. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019, Atlanta, GA, USA, January 29-31, 2019. pp. 279–288. ACM (2019). https://doi.org/10.1145/3287560.3287574
36. Muggleton, S.: Inductive logic programming. New generation computing **8**(4), 295–318 (1991). https://doi.org/10.1007/BF03037089
37. Muggleton, S., Raedt, L.D.: Inductive logic programming: Theory and methods. J. Log. Program. **19/20**, 629–679 (1994). https://doi.org/10.1016/0743-1066(94)90035-3
38. Murakami, Y.: Utilitarian deontic logic. In: Advances in Modal Logic 5, papers from the fifth conference on ”Advances in Modal logic,” held in Manchester, UK, 9-11 September 2004. pp. 211–230. King's College Publications (2004)
39. Paton, H.J.: The categorical imperative: A study in Kant's moral philosophy, vol. 1023. University of Pennsylvania Press (1971)
40. Pereira, L.M., Saptawijaya, A.: Programming Machine Ethics, Studies in Applied Philosophy, Epistemology and Rational Ethics, vol. 26. Springer (2016). https://doi.org/10.1007/978-3-319-29354-7
41. Powers, T.M.: Prospects for a kantian machine. IEEE Intelligent Systems **21**(4), 46–51 (2006)
42. Ray, O.: Hybrid abductive inductive learning. Ph.D. thesis, Imperial College London, UK (2005), `http://ethos.bl.uk/OrderDetails.do?uin=uk.bl.ethos.428111`
43. Reiter, R.: A logic for default reasoning. Artificial intelligence **13**(1-2), 81–132 (1980)
44. Ross, W.D.: The Right and the Good. Oxford University Press, Oxford (1930). https://doi.org/10.2307/2180065
45. Sergot, M.: Prioritised Defeasible Imperatives. Dagstuhl Seminar 16222 Engineering Moral Agents – from Human Morality to Artificial Morality (2016), `https://materials.dagstuhl.de/files/16/16222/16222.MarekSergot.Slides.pdf`, schloss Dagstuhl-Leibniz-Zentrum fuer Informatik
46. Shakerin, F., Gupta, G.: Heuristic based induction of answer set programs: From default theories to combinatorial problems. CoRR **abs/1802.06462** (2018), `http://arxiv.org/abs/1802.06462`
47. Shakerin, F., Salazar, E., Gupta, G.: A new algorithm to automate inductive learning of default theories. TPLP **17**(5-6), 1010–1026 (2017)
48. de Sousa, R.R., Soares, G., D'Antoni, L., Polozov, O., Gulwani, S., Gheyi, R., Suzuki, R., Hartmann, B.: Learning syntactic program transformations from examples. CoRR **abs/1608.09000** (2016), `http://arxiv.org/abs/1608.09000`
49. Srinivasan, A.: The Aleph Manual (version 4). Machine Learning Group, Oxford University Computing Lab (2003), https://www.cs.ox.ac.uk/activities/machlearn/Aleph/aleph.html
50. Varshney, K.R., Alemzadeh, H.: On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. CoRR **abs/1610.01256** (2016), `http://arxiv.org/abs/1610.01256`
51. Wallach, W., Allen, C., Smit, I.: Machine morality: bottom-up and top-down approaches for modelling human moral faculties. AI Soc. **22**(4), 565–582 (2008). https://doi.org/10.1007/s00146-007-0099-0
52. Wexler, R.: When a Computer Program Keeps You in Jail: How Computers are Harming Criminal Justice. NewYork Times (2017), `https://www.nytimes.com/2017/06/13/opinion/how-computers-are-harming-criminal-justice.html`