Learning Answer Set Programming Rules For Ethical Machines

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

[1] Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
Università degli Studi dell'Aquila, Italy
Abeer.Dyoub@graduate.univaq.it,Stefania.Costantini@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. Codes of ethics are abstract rules. These rules are often quite difficult to apply. Abstract principles such as these contain open textured terms that cover a wide range of specific situations. These codes are subject to interpretations and might have different meanings in different contexts. There is an implementation problem from the computational point of view with most of these codes, they lack clear procedures for implementation. In this work we present a new approach based on Answer Set Programming and Inductive logic Programming for monitoring the employees behavior w.r.t. ethical violations of their company's codes of ethics. After briefly reviewing the domain, we introduce our proposed approach, followed by a discussion, then we conclude highlighting possible future directions and potential developments.

1 Introduction

Motivation and Background Machine Ethics is an emerging interdisciplinary field which draws heavily from philosophy and psychology [23]. The Machine Ethics field is concerned with the moral behavior of artificial intelligent agents. Nowadays, with the growing power and increasing autonomy of artificial intelligent agents, which are used in our everyday life performing tasks on our behalf, it has become imperative to equip these agents with capabilities of ethical reasoning. Robots in elder care, robot nannies, virtual companions, chatbots, robotic weapons systems, autonomous cars, etc. are examples of some of the artificial intelligent systems undergoing research and development. These kinds of systems usually need to engage in complex interactions with humans. For this reason, taking into consideration ethical aspects during the design of such machines has become a pressing concern.

The problem of adopting ethical approach to AI has been attracting a lot of attention in the last few years. Lately, the European Commission has published a 'Draft Ethics Guidelines for Trustworthy AI' [18]. In this document, the European Commission's High-Level Expert Group on Artificial Intelligence specifies the requirements of trustworthy AI, and the technical and non technical methods to ensure the implementation of these requirements into the AI system. There is a the world wide urge that ethics should be embedded in the design of intelligent autonomous systems and technologies (IEEE global initiative 'Ethics in Action' [3]). The tech giant 'Google', after a protest from company employees over ethical concerns, ended its involvement in an American Pentagon Project on autonomous weapons [4]. Because of the controversy over its Pentagon work, Google laid down a set of AI principles [5] meant as a guide for future projects. However, the new principles are open to interpretations.

Moral thinking pervades everyday decision making, though understanding the nature of morality and the psychological underpinnings of moral judgment and decision making have been always a big concern for researchers. 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 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 were several attempts for implementing ethical decision making into intelligent autonomous agents using different approaches. But, no fully descriptive and widely accepted model of moral judgment and decision-making exists. None of the developed solutions seems to be fully convincing for providing a trusted moral behavior. In addition, all the existing research in machine ethics try to satisfy certain aspects of ethical decision making but fail to satisfy others. Approaches to machine ethics are classified into: top-down approaches, which try to implement specific normative theory of ethics into the autonomous agent so that 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 in terms of normative theories of ethics. In other words, generalism versus particularism, principles versus case based reasoning. Some researchers argue that morality can only be grounded in particular cases while others defend the existence of general principles related to ethical rules. Both approaches to morality have advantages and disadvantages. We need to adopt a hybrid strategy that allows both top down design and bottom up learning via context-sensitive adaptation of models of ethical behavior.

Contribution Ethics in customer dealings present the company in a good light, and customers will trust the company in the future. Ethics improves the quality of service and fosters positive relationships. Many top leading companies have a booklet called ”code of conduct and ethics” and new employees are made to sign it. However, enforcing codes of conduct and ethics is not an easy task. These codes being mostly abstract and general rules e.g. confidentiality, accountability, honesty, inclusiveness, empathy, fidelity, etc., they are quite difficult to apply. Moreover, abstract principles such as these contain open textured terms ([14]) that cover a wide range of specific situations. They are subject to interpretations and may have different meanings in different contexts. Thus, there is an implementation problem from the computational point of view. It is difficult to use deductive logic to address such a problem ([36], [14]). It is impossible

[3] https://ethicsinaction.ieee.org

[4] https://www.nytimes.com/2019/03/01/business/ethics-artificial-intelligence.html

[5] https://www.blog.google/technology/ai/ai-principles/

for experts to define intermediate rules to cover all possible situations. Codes of ethics in their abstract form are very difficult to apply in real situations [19]. All the above mentioned reasons make learning from cases and generalization crucial for judgment of future cases and violations.

In this work and with the future perspective of ethical chatbots in customer service, we propose an approach to address the problem of evaluating the ethical behavior of customer service employees for violations of the codes of ethics and conduct of their company. Our approach is based on Answer Set Programming (ASP) and Inductive Logic Programming (ILP). We use ASP for ethical knowledge representation and reasoning. The ASP rules needed for reasoning are learned using ILP. ASP, the non-monotonic reasoning paradigm, was chosen because it is common to say that ethical rules are default rules, which means that they tolerate exceptions. This in fact nominates non-monotonic logics which simulate common sense reasoning to be used to formalize 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 and the performance of the available 'solvers', etc. which gained ASP an important role in the field of Artificial Intelligence. ILP was chosen as a machine learning approach because ILP as a logic-based machine learning approach supports two very important and desired aspects of machine ethics implementation into artificial agents viz. explainability and accountability [18], ILP is known for its explanatory power which is compelling: as an action is chosen by the system, clauses of the principle that were instrumental in its selection can be determined and used to formulate an explanation of why a particular action was chosen over others; moreover, ILP also seems better suited than statistical methods to domains in which training examples are scarce as in the case of ethical domain. Many research works have suggested the use of ASP and ILP, but separately, for programming ethical agents which we review in Section five. We think that an approach combining both programming languages would have a great potential for programming ethical agents. Finally we would like to mention that our approach can be applied to generate detailed codes of ethics for any domain.

Structure The Paper is organized as follows: in Sections two, we briefly introduce both ASP and ILP as the logic programming techniques used in this work. In Section three we present our approach with examples. Then Section four we review the research done for modeling ethical agents using ASP and ILP. Then we conclude with future directions in Section five.

2 Background

2.1 ASP Formalism

ASP is a logic programming paradigm under answer set (or "stable model") semantics [16], which applies ideas of autoepistemic logic and default logic. ASP features a highly declarative and expressive programming language, oriented towards difficult search problems. It has been used in a wide variety of applications in different areas like problem solving, configuration, information integration, security analysis, agent systems, semantic web, and planning. ASP has emerged from interaction between two lines of research: first on the semantics of negation in logic programming, and second on applications of satisfiability solvers to search problems [21]. 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. The expressiveness of ASP, the readability of its code and the performance of the available ”solvers” gained ASP an important role in the field of artificial intelligence.

An answer set Program is a collection of rules of the form,

$$H \leftarrow\ A_1, \ldots, A_m, not\, A_{m+1}, \ldots, not\, A_n$$

were each of A_i's is a literal in the sense of classical logic. Intuitively the above rule means that if $A_1, \ldots, A_m$ are true and if $A_{m+1}, \ldots, A_n$ 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 *unit rule*, or *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, each answer set is seen as a solution of given problem, encoded as an ASP program (or, better, the solution is extracted from an answer set by ignoring irrelevant details and possibly re-organizing the presentation). 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. For more information about ASP and its applications the reader can refer, among many, [15], [12] and the references therein field of artificial intelligence.

2.2 ILP Approach

ILP [24] is a branch of artificial intelligence (AI) which investigates the inductive construction of logical theories from examples and background knowledge. It is the intersection between logic programming and machine learning. From computational logic, inductive logic programming inherits its representational formalism, its semantical orientation, and various well-established techniques.

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* [25]. 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. An asterisk $*$ denotes an arbitrary recall. A scheme can contain special *placemaker* terms of the form $\sharp$ *type*, *+type* and *-type*, which stand, respectively, for ground terms, input terms and output terms of a predicate *type*. Each set *M* of mode declarations is associated with a set of clauses $\mathcal{L}(M)$, called the language of *M*, such that $C = a \leftarrow l_1, \ldots, l_n \in \mathcal{L}(M)$ iff the head atom *a* (resp. each body literal l_i) is obtained from some head (resp. body) declaration in *M* by replacing all $\sharp$ *placemakers* with ground terms and all + (resp. -) *placemakers* with input (resp. output) variables. 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 [26].

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 learned knowledge. Most of the work on ILP frameworks has focused on learning definite logic programs (e.g. among many, [24], [35]) and normal logic programs (e.g. [11]). In the recent years, several new learning frameworks and algorithms have been introduced for learning under the answer set semantics. In fact generalizing ILP to learn ASP makes ILP more powerful. Among many, refer to [32]. [22], [30], [34], and [20].

3 Our Approach: An Application

Codes of ethics in domains such as customer service are mostly abstract general codes, which make them quite difficult to apply. Examples, confidentiality, accountability, honesty, fidelity, etc. They are subject to interpretations and may have different meanings in different contexts. Therefore it is quite difficult if not impossible to define codes in a manner that they maybe applied deductively. There are no intermediate rules that elaborate the abstract rules or explain how they apply in concrete circumstances. Consider for example the following codes of ethics taken from a customer service code of ethics and conduct document of some company:

Confidentiality: The identity of the customer and the information provided will be shared only on a "need-to-know" basis with those responsible for addressing and resolving the concern.

Accuracy: We shall do all it can to collect, rely and process customer requests and complaints accurately. We shall ensure all correspondence is easy to understand, professional and accurate.

Accountability: Our employees are committed to own a service request or a complaint received and they are responsible for finding answers and getting the issue resolved. If the employee cannot solve the problem himself, he is expected to find someone who can and follow up until the issue is resolved.

Abstract principles such as these seems reasonable and appropriate, but in fact it is very hard to apply them in real-world situations [19] (e.g. how can we precisely define ”We shall ensure all correspondence is easy to understand, professional and accurate.”? or ”shall do all it can to collect, rely and process customer request and complaint accurately.”?). It is not possible for experts to define intermediate rules to cover all possible situations to which a particular code applies. In addition, there are many situations in which obligations might conflict. An important question to ask here is how can the company's managers evaluate the ethical behavior of employees in such setting. To achieve this end, and help managers to have detailed rules in place for monitoring the behavior of their employees at customer service for violations of the company's ethical codes, we propose an approach for generating these detailed rules of evaluation from interactions with customers. So, the new codes of ethics to be used for ethical evaluation are a combination of the existing clear codes (those that give a clear evaluation procedure that

can be deductively encoded using ASP) and the newly generated ones. The approach uses ASP Language as the knowledge representation and reasoning language. ASP is used to represent the domain knowledge, the ontology of the domain, and scenarios information. Rules required for ethical reasoning and evaluation of the agent behavior in a certain scenario are learned using XHAIL [30], which is a Non-monotonic ILP algorithm. The inputs to the system are a series of scenarios(cases) in the form of requests and answers, along with the ethical evaluation of the response considering each particular situation. The system remembers the facts about the narratives and the annotations given to it by the user, and learns to form rules and relations that are consistent with the evaluation given by the user of the responses to the given requests.
To illustrate our approach, let us consider the following scenario: a customer contacting the customer service asking for a particular product of the company, and the employee talking about the product characteristics and trying to convince the customer to buy the product. (S)he started saying that the product is environmentally friendly (which is irrelevant in this case), and this is an advantage of their product over the same products of other companies. The question: is it ethical for the employee to say that? The answer is no, it is unethical to make use of irrelevant but sensitive slogans like environmentally friendly" to attract and provoke the customers to buy a certain product or service. This would be a violation of 'Honesty'.
We can form an ILP task $ILP(B, E = \{E^+, E^-\}, M)$ for our example, where B is the background knowledge:

$$B = \begin{cases} ask(customer, infoabout(productx)). \\ answer(environmentallyFriendly). \\ sensitiveSlogan(environmentallyFriendly). \\ not\_relevant(environmentallyFriendly). \\ answer(xxx). \quad sensitiveSlogan(xxx). \quad not\_relevant(xxx). \\ answer(yyy). \quad sensitiveSlogan(yyy). \quad not\_relevant(yyy). \\ answer(zzz). \quad not\_sensitiveSlogan(zzz). \quad relevant(zzz). \\ answer(eee). \quad not\_sensitiveSlogan(eee). \quad relevant(eee). \\ not\_relevant(X) :- not \quad relevant(X), answer(X). \\ not\_sensitiveSlogan(X) :- not \quad sensitiveSlogan(X), answer(X). \end{cases}$$

E are the positive and negative examples:

$$E^+ = \begin{cases} example \quad unethical(environmentallyFriendly). \\ example \quad unethical(xxx). \\ example \quad unethical(yyy). \end{cases}$$

$$E^- = \begin{cases} example \quad notunethical(zzz). \\ example \quad notunethical(eee). \end{cases}$$

M is The mode declarations:

$$M = \begin{cases} modeh \quad unethical(+answer). \\ modeb \quad sensitiveSlogan(+answer). \\ modeb \quad notsensetiveSlogan(+answer). \\ modeb \quad notrelevant(+answer). \\ modeb \quad relevant(+answer). \end{cases}$$

In the running example, E contains three positive examples and two negative examples which must all be explained. XHAIL derives the hypothesis in three steps process:
Step 1: The Abductive Phase: the head atoms of each Kernel Set are computed. The set of abducibles (ground atoms) is $\varDelta = \bigcup_{i=1}^{n} \alpha_i$ such that $B \bigcup \varDelta \models E$ where each α_i is a ground instance of the *modeh(d)* declaration atom. This is a straight-forward abductive task. For our example there is only one *modeh* declaration. Then the set $\varDelta$ contain ground instances of this atom in the single *modeh* declaration. So the set of abducibles $\varDelta$ for our example would be:

$$\varDelta = \begin{cases} unethical(environmentallyFriendly). \\ unethical(xxx). \\ unethical(yyy). \end{cases}$$

Step 2: The Deductive Phase: This step computes the body literals of a Kernel Set. i.e., the clause $\alpha_i \leftarrow \delta_i^1 \dots \delta_i^{m^i}$ for each $\alpha_i \in \varDelta$ is computed, where $B \bigcup \varDelta \models \delta_i^j, \forall 1 \leq i \leq n, 1 \leq j \leq m_i$ and each clause $\alpha_i \leftarrow \delta_i^1 \dots \delta_i^{m^i}$ is a ground instance of a rule in $\mathcal{L}(M)$ (the language of *M*, where *M* is the mode declarations). To do this, each head atom is saturated with body literals using a nonmonotonic generalization of the Progol level saturation method ([25]).In our example, $\varDelta$ contains three atoms where each one leads to a clause k_i, so, we will have K_1, K_2, K_3. The first atom $\alpha_1 = unethical(environmentallyFriendly)$ is initialized to the head of the clause K_1. The body of K_1 is saturated by adding all possible ground instances of the literals in *modeb(s)* declarations that satisfy the constraints mentioned earlier. There are ten ground instances of the literals in the *modeb(d)* declarations, but only two of them, i.e. $sensitiveSlogan(environmentallyFriendly)$ and $not\_relevant(environmentallyFriendly)$ can be added to the body of K_1. At the end of the deductive phase we will have the set of ground clauses K:

$$K = \begin{cases} K1 = unethical(environmentallyFriendly) \leftarrow \\ \qquad\qquad sensitiveSlogan(environmentallyFriendly), \\ \qquad\qquad not\_relevant(environmentallyFriendly). \\ K2 = unethical(xxx) \leftarrow sensitiveSlogan(xxx), not\_relevant(xxx). \\ K3 = unethical(yyy) \leftarrow sensitiveSlogan(yyy), not\_relevant(yyy). \end{cases}$$

and the set of their "variablized" version that is obtained by replacing all input and output terms by variables:

$$K_v = \begin{cases} unethical(V) \leftarrow sensitiveSlogan(V), not\_relevant(V). \\ unethical(V) \leftarrow sensitiveSlogan(V), not\_relevant(V). \\ unethical(V) \leftarrow sensitiveSlogan(V), not\_relevant(V). \end{cases}$$

Step 3: The Inductive Phase: By construction, the Kernel Set covers the provided examples. In this phase XHAIL computes a compressive theory $H = \bigcup_{i=1}^{n'} \alpha_i \leftarrow d_i^1, \ldots, d_i^{m'_i}$ that subsumes K and entails E w.r.t. B. This is done through actual search for hypothesis which is biased by minimality i.e. preference towards hypothesis with fewer literals. Thus a hypothesis is constructed by deleting from K_v as many literals (and clauses) as possible while ensuring correct coverage of the examples. This is done by subjecting K_v to syntactic transformation of its clauses which involves two new predicates $try/3$ and $use/2$. This syntactic transformation results in the following defeasible program:

$$U_{K_v} = \begin{cases} unethical(V) \leftarrow use(1,0), try(1,1,vars(V)), try(1,2,vars(V)). \\ try(1,1,vars(V)) \leftarrow use(1,1), sensitiveSlogan(V). \\ try(1,1,vars(V)) \leftarrow not \quad use(1,1). \\ try(1,2,vars(V)) \leftarrow use(1,2), not\_relevant(V). \\ try(1,2,vars(V)) \leftarrow not \quad use(1,2). \end{cases}$$

literals and clauses necessary to cover the examples are selected from U_{K_v} by means of abducing a set of $use/2$ atoms as explanation for the examples from the ALP (Abductive Logic Programming) task $ALP(B \cup U_{K_v}, \{use/2\}, E)$. $\Delta_2 = \{use(1,0), use(1,1), use(1,2)\}$ is a minimal explanation for this ALP task. $use(1,0)$ is the head atom of one of the K_v clauses (which are identical in this example), $use(1,1)$ and $use(1,2)$ correspond to the body literals. The output hypothesis is constructed by these literals. The three clauses in K_v produce identical transformations resulting in the same final hypothesis:

$$H = \Big\{ unethical(V) \leftarrow sensitiveSlogan(V), not\_relevant(V), answer(V).$$

XHAIL did learn this rule in a total time of 1.671 seconds on AMD Athlon(tm) II Dual-Core M300x2 laptop PC running Ubuntu 14.04 with 3.6G Ram: loading time : 0.987s, abduction : 0.221s, deduction : 0.031s, induction : 0.055s

Let us now consider our agent having three cases together, the above mentioned case and the following two cases(scenarios) along with a set of examples for each case.
case1: an employee give information about client1 to client2 without checking or being sure that client2 is authorized to be given such information. This behavior is unethical because it violates 'Confidentiality' which is very critical especially when dealing with sensitive products and services like services or products provided to patients with critical medical conditions.
case2: a customer contacting customer service asking to buy a certain product x. In this context the customer asks about a similar product of another competitor company which

is slightly cheaper. Then the employee, in order to convince the customer to buy their product and not think about the other company product, said that the other company uses substandard materials in their production. The question: is this an ethical answer from the employee to say that the other company uses substandard materials, supposing that it is true? The answer: no. In general, it is true that the employee should be truthful with the customer, but in this context, the answer is not ethical because it is not ethical and not professional to talk bad about other competitor companies.

From these three cases our agent learned the following three rules for evaluating the employees ethical behavior (for the lack of space we omitted the details):

$$H = \begin{cases} unethical(V) \leftarrow sensitiveSlogan(V), not\_relevant(V), answer(V). \\ unethical(giveinfo(V1, V2)) \leftarrow \\ \qquad context(competitor(V2)), badinfo(V1), info(V1), company(V2). \\ unethical(tell(V2, infoabout(V2))) \leftarrow \\ \qquad not\_authorized(tell(V1, infoabout(V2))), client(V1), client(V2). \end{cases}$$

The above three hypotheses were learned by our agent in a total time of 9.391 seconds: loading time : 0.271s, abduction : 0.124s, deduction : 0.091s, induction : 8.809s . In addition, supposing that our agent already have the following rule as a background knowledge in his knowledge base:

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

which says that it is unethical to give incorrect information to the customers. So now our agent have four rules for ethical evaluation (the one that she already have plus the three learned ones).

4 Related Work

Engineering machine ethics (building practical ethical machines) is not just about traditional engineering, we need to find out how to practically build machines that are ethically constrained and also reason about ethics, which of course involve philosophical aspects. Even though it is more computational by nature. Below we review research works which used ASP for modeling ethical agents and then those that use ILP.

4.1 Non-monotonic Logic and Ethical Reasoning

Ethical reasoning is a form of common sense reasoning. Thus, it seems appropriate to use non-monotonic logics which simulates common sense reasoning to formalize different ethical conceptions. Moreover, logical representations help to make ideas clear and highlight differences between different ethical systems. Ethical rules usually dictate the ethical behavior, i.e. help us to decide what to do and what not to do. Thus, to achieve this ethical behavior, it is required to define a decision making procedure. ASP as a purely declarative nonmonotonic logic paradigm has been nominated as a modern logic-based AI technique to model ethical reasoning systems. Using the nonmonotonic logic of ASP offers a more feasible approach than the deontic logics approaches (like [9] and [28]), since it can address not only the consequentialist ethical systems but also deontic ones as it can represent (limited forms of) modal logic and deontic logics. In addition, the existence of 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.

Using nonmonotonic logic is appropriate to address the opposition between generalism and particularism by capturing justified exceptions in general ethics rules. This opposition corresponds to the old opposition between written laws and the cases on which the laws are based. General rules that they may be correct in theory, but not applicable to all particular cases. In [13], the authors formalized three ethical conceptions (the Aristotelian rules, Kantian categorical imperative, and Constant's objection) using nonmonotonic logic, particularly ASP. Each model is illustrated using the classical dilemma of lying [13]. In the case of lying, default rules with justified exceptions could be used to satisfy a general rule that prohibit lying, while simultaneously recommending telling a lie in given particular situations where the truth would violate other rules of duty.

[8] proposes an ethical modular architecture that allows for systematic and adaptable representation of ethical principles. This work is implemented in ASP. In their framework, the authors model the knowledge of the world in separated models from those used for ethical reasoning and judgment. Many theories of the right were modeled in this paper. However, as mentioned before the framework presented in this paper assesses the permissibility of an action or a set of actions using different theories of Good and Right separately, i.e. it only permits to judge an option with respect to a single ethical principle. It doesn't handle the conflicting decisions given by different theories, i.e. doesn't provide a final decision for the agent about what it should do as a result.

In the context of logic-based ethics, 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. [27]. Their formalization embeds the moral requirements directly into the model of a situation. By indicating, e.g., whether a killing is intentional or not, the program is told whether the outcome of the action fits with the ethical rules in place. This approach fails in representing the actual reasoning that underlies moral decision making, namely, what constitutes intentionality. Furthermore, because they automatically specify the ethical character of the situation outcome, one needs to write different programs for each case. This is redundant and can lead to inconsistencies. Their approach was also criticized by [7], for there is no account for causality and ethical responsibility because action and its consequences are not dynamically linked; the relationship between them is stated by the programmer rather than inferred. In addition, it fails to provide a general framework to model morality computationally because the model cannot logically confront ethical theories making their assumptions explicit and cannot enable us to explore and generate new ethical dilemmas for further testing.

In [10], the authors introduce a generic judgment model that an agent can use in order to judge the ethical dimensions of both its own behavior and the other agents' behaviors. This model is based on a rationalist and explicit approach that distinguish theory of good and theory of right. A proof of concept was implemented in ASP. However, the model is still based on a qualitative approach. Whereas we 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.

In [33], Sergot provides an alternative representation to the argumentative representation of a moral dilemma case concerning a group of diabetic persons, presented in [5], 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 [17]. The proposed solution was implemented in ASP.

4.2 ILP and Machine Ethics

As mentioned above, ethics is more complicated than following a single absolute ethical principle. Thus, according to Ross ([31]), any single-principled ethical theory like Act Utilitarianism is sentenced to fail. Ross suggested that ethical decision making involves considering several Prima Facie duties (duties that in general we should try to follow, where on some occasions the strongest duty can override others). Ross' Theory of Prima Facie Duties seems to more completely account for the different types of ethical obligations that most of us recognize, than any single-principled ethical theory. However, Ross' Theory gives us no decision procedure for determining which duty becomes the strongest one, when several duties pull in different directions as often happens in an ethical dilemma. Rawls' "Reflective Equilibrium" approach [29] was suggested later for reflecting on duties wherever necessary, in order to achieve an acceptable coherence amongst them, the so-called "equilibrium" which serves as decision procedure, lacking in Ross' theory. ILP was used to handle the non-classical relationships that might exist between different duties.

In [3], authors created a system called W.D. Their system follow the theory of prima facie duties of Ross [31]. In W.D., the strength of each duty is measured by assigning it a weight, capturing the view that a duty may take precedence over another. W.D. computes, for each possible action, the weighted sum of duty satisfaction, and returns the greatest sum as the right action. In order to improve the decision, in the sense of conforming to a consensus of correct ethical behavior, the weight of a duty is allowed to be adjusted through a supervised learning, by acquiring suggested action from the user. This weight adjustment to refine moral decisions is inspired by the reflective equilibrium of Rawls [29]. W.D. uses inductive logic programming [24] to achieve this end. W.D. uses ILP to learn the relation *supersedes(A1,A2)* which states that action *A1* is preferred over action *A2* in an ethical dilemma involving these choices.

MedEthEx [4], and EthEl [1] are two systems based on a more specific theory of prima facie duties viz., the principle of Biomedical ethics of Beauchamp and Childress [6]. Moreover the two systems are implemented using ILP [24]. ILP is used in both MedEthEx, and EthEl to learn the relation *supersedes(A1, A2)*, i.e., whether action A1 supersedes (i.e., is ethically preferable to) action A2. The training (positive) examples comprise cases, where each case is associated with an estimate satisfaction/violation value of each duty for each possible action (scaled from -2 to 2) and the ethically preferred action for the case. The considered cases are a variety of the following type of ethical dilemma: ”A healthcare professional has recommended a particular treatment for her competent adult patient, but the patient has rejected it. Should the healthcare professional try to change the patient’s mind or accept the patient’s decision as final?” 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 (General Ethical Dilemma Analyzer)[2] is another System that uses ILP as a machine learning technique to discern ethical principles that resolve ethical dilemmas due to conflicting obligations and duties. GenEth has been used to codify principles in a number of domains pertinent to the behavior of autonomous systems.

5 Conclusions and Future Directions

In this paper we reviewed approaches to modeling ethics using ASP (rule-based approaches) and ILP (case-based learning approaches). Then we presented an approach that makes use of ASP for ethical knowledge representation and reasoning, and uses inductive logic programming for learning ASP rules needed for ethical reasoning. Combining ASP with ILP for modeling ethical agents provides many advantages: increases the reasoning capability of our agent; promotes the adoption of hybrid strategy that allow both topdown 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 which equips our agent with the capacity to give an ethical evaluation and explain the reasons behind this evaluation. In other words, our method supports transparency and accountability of such models, which facilitates instilling confidence and trust in our agent. Furthermore, in our opinion and for the sake of transparency, evaluating the ethical behavior of others should be guided by explicit ethical rules determined by competent judges or ethicists or through consensus of ethicists. Our approach provides support for developing these ethical rules.

Computational techniques, such as neural networks, can be viewed as making use of a form of inductive inference. However, ILP algorithms, unlike neural networks, output rules which are easily understood by people. Statistical machine learning methods produce models that are not comprehensible for humans because they are algebraic solutions to optimization problems such as risk minimization or data likelihood maximization. These methods do not produce any intuitive description of the learned model. Lack of intuitive descriptions makes it hard for users to understand and verify the underlying rules that govern the model. Also, these methods cannot produce a justification for a prediction they compute for a new data sample. Furthermore, if prior knowledge (background knowledge) is extended in these methods, then the entire model needs to be re-learned. Finally, no distinction is made between exceptions and noisy data in these methods. This makes ILP particularly appropriate for scientific theory formation tasks in which the comprehensibility of the generated knowledge is essential. Moreover, in an ill-defined domain like the ethics domain, it is infeasible to define abstract codes in precise and complete enough terms to be able to use deductive problem solvers to apply them correctly. A combination of deductive (rule-based) and inductive (case-based learning) is needed. The use of ASP allows us to encode the domain information plus the nonmonotonic domain rules that are already available and can help our agent in the evaluation process. However, in the many other cases where we don't have clear intermediate rules for evaluation, learning is needed to learn these rules and then add them to our knowledge base to be used for future evaluations.

With respect to the approach of the works mentioned in the previous section, the authors used ILP to learn rules to help decide between two or more available actions based on a set of involved ethical duties. So their approach can be applied to choose the most ethical action when we have specific clear ethical duties involved and to do so we need to assign weights of importance(priority) to these duties for each available action, then the system computes the weighted sum for each action, and the one with highest weighted sum is the best action to do. In this approach it is not really clear the basis of assigning weights to duties(we doubt whether we can really quantify the importance of ethical duties on a grade from 2 to -2 as was done in these works). On the other hand, in our approach we use ILP to generate rules for ethical evaluation of actions(in the case of the application we are handling in this paper, actions are the responses to requests from customers) based on different facts extracted from cases. In other words ILP is used to learn the relation between the evaluation of an action to be ethical or unethical and the related facts in the case scenario. To this end, different facts are extracted from the case scenario and our system try to find the relation between these facts and the conclusion (ethical or un ethical or probably unknown).our approach can be used to generate ethical rules to follow when there is no ethical rules available in place for evaluation, by considering the involved facts and possibly involving counterfactual reasoning in the evaluation. We think that our approach is more general and can be used to generate ethical rules for any domain (and/or elaborate existing ones).

As a matter of fact XHAIL provides an appropriate framework for learning ethical rules for customer service. However XHAIL has the following limitations: the problem of scalability: The computation of a hypothesis H depends on the Kernel Set (K) generation, particularly on the choice of Δ (the set of heads of K's clauses), which is a set of instances of head mode declaration atoms derived from B; the Kernel Set is generated from all positive examples at once, then XHAIL performs a search in the space of theories that subsume it, in order to arrive to a "good" hypothesis. Thus, when the size of the examples is small, XHAIL performs well. But with the increasing size of examples space, XHAIL scales poorly, two reasons are behind this. First: the increased computational complexity of abduction, which lies at the core of its functionality; second: the combinatorial complexity of learning whole theories which may result in an intractable search space. Furthermore, every time we want to add new cases, XHAIL need to relearn the new hypothesis from the whole set of examples (old ones plus the new added ones). Therefore, to cope with large volumes of sequential data and also to cope with ethics change over time, we need an incremental learning technique that is able to revise the old learned hypothesis when a new set of examples arrive. In fact we are working now to improve the ethical evaluation capabilities of our agent by using an incremental learning algorithm like ILED ([20]) to overcome the limitations mentioned above. So our agent can learn incrementally from the interactions with customers to give more accurate evaluations to customer service employees ethical behavior. Furthermore, we would like to test our agent in a real chat scenario. Finally, as another future direction we would like to investigate the possibility of judging the ethical behavior from a series of related chat sessions.

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-08-02, 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), `http://www.aaai.org/Library/AAAI/aaai14contents.php`
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. 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)
6. Beauchamp, T.L., Childless, J.F.: Principles of biomedical ethics. International Clinical Psychopharmacology **6**(2), 129–130 (1991)
7. Berreby, F., Bourgne, G., Ganascia, J.G.: Modelling moral reasoning and ethical responsibility with logic programming. In: Logic for Programming, Artificial Intelligence, and Reasoning. pp. 532–548. Springer (2015)
8. Berreby, F., Bourgne, G., Ganascia, J.: A declarative modular framework for representing and applying ethical principles. In: AAMAS. pp. 96–104. ACM (2017)
9. 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`
10. 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)
11. 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)
12. Dyoub, A., Costantini, S., Gasperis, G.D.: Answer set programming and agents. Knowledge Eng. Review **33**, e19 (2018)
13. Ganascia, J.G.: Modelling ethical rules of lying with answer set programming. Ethics and information technology **9**(1), 39–47 (2007)

14. Gardner, A.v.d.L.: An artificial intelligence approach to legal reasoning. MIT Press (1987)
15. Gelfond, M.: Answer sets. In: Handbook of Knowledge Representation. Chapter 7, Foundations of Artificial Intelligence, vol. 3, pp. 285–316. Elsevier (2007)
16. 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)
17. Hansen, J.: Prioritized conditional imperatives: problems and a new proposal. Autonomous Agents and Multi-Agent Systems **17**(1), 11–35 (2008)
18. High-Level Expert Group on Artificial Intelligence: Draft ethics guidelines for trustworthy AI. European Commission, Brussel (2018), https://ec.europa.eu/digital-single-market/en/news/draft-ethics-guidelines-trustworthy-ai
19. Jonsen, A.R., Toulmin, S.E.: The abuse of casuistry: A history of moral reasoning. Berkeley: Univ of California Press (1988)
20. 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
21. Kautz, H.A., Selman, B., et al.: Planning as satisfiability. In: ECAI. vol. 92, pp. 359–363. Citeseer (1992)
22. Law, M., Russo, A., Broda, K.: Iterative learning of answer set programs from context dependent examples. TPLP **16**(5-6), 834–848 (2016)
23. Moor, J.H.: The nature, importance, and difficulty of machine ethics. IEEE intelligent systems **21**(4), 18–21 (2006)
24. Muggleton, S.: Inductive logic programming. New generation computing **8**(4), 295–318 (1991)
25. Muggleton, S.: Inverse entailment and progol. New Generation Comput. **13**(3&4), 245–286 (1995), https://doi.org/10.1007/BF03037227
26. 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
27. 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
28. Powers, T.M.: Prospects for a kantian machine. IEEE Intelligent Systems **21**(4), 46–51 (2006)
29. Rawls, J.: A theory of justice. Harvard University Press, Cambridge (1971)
30. Ray, O.: Nonmonotonic abductive inductive learning. J. Applied Logic **7**(3), 329–340 (2009), https://doi.org/10.1016/j.jal.2008.10.007
31. Ross, W.D.: The Right and the Good. Oxford University Press, Oxford (1930)
32. Sakama, C., Inoue, K.: Brave induction: a logical framework for learning from incomplete information. Machine Learning **76**(1), 3–35 (2009)
33. 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
34. Shakerin, F., Salazar, E., Gupta, G.: A new algorithm to automate inductive learning of default theories. TPLP **17**(5-6), 1010–1026 (2017)
35. 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
36. Toulmin, S.E.: The uses of argument. Cambridge university press (2003)