Commonsense Reasoning meets Theorem Proving

Ulrich Furbach and Claudia Schon*
{uli,schon}@uni-koblenz.de

Universität Koblenz-Landau

Abstract. The area of commonsense reasoning aims at the creation of systems able to simulate the human way of rational thinking. This paper describes the use of automated reasoning methods for tackling commonsense reasoning benchmarks. For this we use a benchmark suite introduced in literature. Our goal is to use general purpose background knowledge without domain specific hand coding of axioms, such that the approach and the result can be used as well for other domains in mathematics and science. Furthermore, we discuss the modeling of normative statements in commonsense reasoning and in robot ethics.[1]

1 Introduction

Commonsense reasoning aims at creating systems able to simulate the human way of rational thinking. This area is characterized by ambiguity and uncertainty since it deals with problems humans are confronted with in everyday life. Humans performing reasoning in everyday life do not necessarily obey the rules of classical logic. This causes humans to be susceptible to logical fallacies but on the other hand to draw useful conclusions automated reasoning systems are incapable of. Humans naturally reason in the presence of incomplete and inconsistent knowledge, are able to reason in the presence of norms as well as conflicting norms, and are able to quickly reconsider their conclusions when being confronted with additional information. The versatility of human reasoning illustrates that any attempt to model the way humans perform commonsense reasoning has to use a combination of many different techniques. Such techniques can also be subsumed under the keyword 'cognitive computing' which was coined by IBM after the success of their Watson-System in the Jeopardy! quiz show.

In this paper we describe the progress we made so far in creating a system able to tackle commonsense reasoning benchmarks. We start in Section 2 with a short description of a natural language question answering project, from which we learned a lot about implementing commonsense reasoning. Namely to combine and to apply techniques from automated theorem proving, natural language processing, large ontologies as background knowledge and machine learning.

* Work supported by DFG FU 263/15-1 'Ratiolog'

[1] This paper is an extended version of the informal proceedings [9] and [10].

In Section 3 we discuss the use of automated reasoning in cognitive computing. We introduce several benchmarks for the area of commonsense reasoning and describe the techniques which we combine therein. In order to find the more plausible answer to the benchmark problems, machine learning techniques come to use and we present first experimental results from this area. In a final Section 4 we discuss the modeling of normative rules in commonsense reasoning and in robot ethics.

2 Automated Reasoning in the Question Answering system LogAnswer

This section introduces the Loganswer project, which was finished only recently and from which we learned some valuable lessons for the topic of this paper. This project [6] researched a system for open domain question answering from a snapshot of the German Wikipedia. The user enters a question in natural language and the LogAnswer system provides answers together with highlighted textual sources. Opposed to other question answering systems, the LogAnswer system does not rely solely on shallow linguistic methods but uses the automated theorem prover Hyper to compute the answers. For this process possible answer candidates are determined by syntactic keyword search and ranked by machine learning techniques. For the 200 best answer candidates from this process Hyper is invoked with the semantic contents of the answer candidate represented in predicate logic together with the query and background knowledge. The whole process of answering a question is time critical, since users are not willing to accept slow respond times. It is not surprising that Hyper is not always able to construct a proof when confronted with an answer candidate for the question under consideration in reasonable time. If no proof can be found within a certain time limit, a technique called relaxation is used. This techniques allows to weaken or drop subgoals of the question in order to enable Hyper to find a proof within the time limit. Of course, this technique comes at the expense of accuracy. In the last step, all proofs are ranked by machine learning techniques and for the three best proofs a natural language answer is presented to the user.

One finding which can be seen as an especially interesting insight from the LogAnswer project is the following fact: When combined with suitable background knowledge and machine learning techniques, automated theorem provers can be successfully applied even in domains where exact yes or no answers cannot be expected.

In the sequel we suggest to use first-order logic automated reasoning techniques to tackle commonsense reasoning tasks. It is reasonable to question the choice of first-order logic since there are many other logics, like defeasible logic and other non-monotomic logics, which seem to be a much better choice. Using first-order logic, however, has the crucial advantage that it allows the usage of highly optimized theorem provers. In our case we use the Hyper theorem prover ([2]) to solve the reasoning tasks occurring in commonsense reasoning problems. At first glance it seems to be a drawback of first-order logic theorem provers

```
1: My body cast a shadow over the grass. What was the CAUSE of this?

 1. The sun was rising.
 2. The grass was cut.

13: The pond froze over for the winter. What happened as a RESULT?

 1. People skated on the pond.
 2. People brought boats to the pond.
```

Fig. 1: Example problems 1 and 13 from the COPA challenge.

that they are only able to answer yes or no (or sometimes even don't answer at all). However, our experiences in the LogAnswer project demonstrated that the combination of background knowledge, a theorem prover with its proof objects like (partial) proofs or (partial) models and machine learning techniques is able to come to impressive conclusions. We are convinced that the combination of the afore described techniques leads to a result which is much more than the sum of its parts.

3 Automated Reasoning in Cognitive Computing

The previous section demonstrated that for natural language question answering several different techniques and knowledge sources have to be combined in a cooperative and efficient manner. This fits nicely under the term 'cognitive computing' which was coined by IBM research in order to describe such 'Watson-like' systems ([5]). We consider commonsense reasoning as an area which perfectly fits the prerequisites to be tackled by cognitive computing.

3.1 Benchmarks for Commonsense Reasoning

For a long time, no benchmarks in the field of commonsense reasoning were available and most approaches were tested only using small toy examples. Recently, this problem was remedied with the proposal of various sets of benchmark problems. There is the Winograd Schema Challenge [17] whose problems have a clear focus on natural language processing whereas background knowledge has an inferior standing. Another example is the Choice Of Plausible Alternatives (COPA) challenge[2] [26] consisting of 1000 problems equally split into a development and a test set. Each problem consists of a natural language sentence describing a scenario and a question. In addition to that two answers are provided in natural language. The task is to determine which one of these alternatives is the most plausible one. Figure 1 presents two problems from this benchmark suite. Like in the two presented examples, the questions always ask either for the *cause* or the *result* of an observation.

[2] Available at http://people.ict.usc.edu/∼gordon/downloads/COPA-questions-dev.txt.

`The triangle opened the door, stepped outside and started to shake. Why did the triangle start to shake?`
$exit(e1, lt) \wedge shake(e2, lt) \wedge seq(e1, e2)$

1. `The triangle is upset.` $unhappy(e3, lt)$
2. `The triangle is cold.` $cold(e4, lt)$

Fig. 2: Narrative and formalization of an example problem no. 44 from the Triangle-COPA challenge.

Even though for the COPA challenge capabilities for handling natural language are necessary, background knowledge and commonsense reasoning skills are crucial to tackle these problems as well, making them very interesting to evaluate cognitive systems. All existing systems tackling the COPA benchmarks focus on linguistic and statistical approaches by calculating correlational statistics on words.

Another set of benchmarks is the Triangle-COPA challenge[3] [19]. This is a suite of one hundred logic-based commonsense reasoning problems which was developed specifically for the purpose of advancing new logical reasoning approaches. The structure of the problems is the same as in the COPA Challenge, however, the problems in the Triangle-COPA challenge are not only given in natural language but also in first-order logic.

Figure 2 depicts a problem from this benchmark suite consisting of a natural language description as well as first-order logic representation of a situation, a question, and two alternative answers.

Until now only one logic based system is able to tackle the Triangle-COPA benchmarks: [19] and more recently [11] use abduction together with a set of hand-coded axioms. Furthermore, there is a preliminary approach using deontic logic to address the problems given in these benchmarks [7]. We refrain from using hand-coded knowledge and suggest to use knowledge bases containing commonsense knowledge like OpenCyc [16], SUMO [24, 25], ConceptNet [18] and Yago [27] together with a theorem prover instead.

3.2 Combination of Techniques

As described afore, the creation of a system for commonsense reasoning requires cognitive computing, in particular the combination of techniques from different areas. Even gathering appropriate background knowledge for a specific benchmark problem requires the use of different techniques. Figure 3 depicts the different steps necessary to gather suitable background knowledge for a given COPA problem. When combining an example problem with background knowledge, several problems have to be solved:

1. If the problem is given in natural language it has to be transformed into a logical representation.

[3] Available at https://github.com/asgordon/TriangleCOPA/.

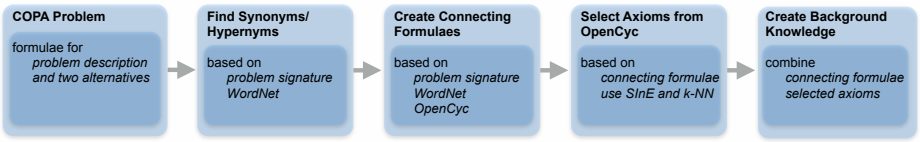

Fig. 3: Gathering background knowledge for a benchmark problem. The starting formulae are generated by transforming the natural language problems of COPA into first-order logic using the Boxer system.

2. The predicate symbols used in the formalization of the example are unlikely to coincide with the predicate symbols used in the background knowledge.
3. The background knowledge is too large to be considered as a whole.

The first problem can be solved using the Boxer [4] system which is able to transform natural language into first-order logic formulae. We assume that this is done before the techniques given in Figure 3 are applied. Please note that this step is not necessary when benchmarks given in first-order logic are considered, like it is the case for the Triangle-COPA challenge.

We address the second problem by using WordNet [20] to find synonyms and hypernyms of the predicate symbols used in the formalization of the example. Note that the formalization of the example consists both of the formulae describing the situation as well as the formulae for the two alternatives. In the next step, predicate symbols used in OpenCyc [16], which are similar to these synonyms and hypernyms are determined. With the help of this information a connecting set of formulae is created. In this step, it is also necessary to adjust the arity of predicate symbols which is likely to differ, since Boxer only creates formulae with unary or binary predicates.

The third problem is addressed using selection methods. For this, all predicate symbols occurring in the formalization of the example and in the connecting set of formulae are used. As selection methods, SInE as well as k-NN as they are implemented in the E.T. metasystem [15] come to use. The selected axioms are combined with the connecting set of formulae and the resulting set of formulae constitutes the background knowledge for the example at hand.

The COPA challenge contains two different categories of problems. In the first category, a sentence describing an observation is given and it is asked for the *cause* of this observation. Probem no. 1 given in Figure 1 is an example for a question in this category. In this case, the task is to determine which of the two provided alternatives is more likely to be the cause of the observation described in the sentence. We call this category the *cause category*.

In the other category a sentence describing an observation is given and it is asked about the *result* of this observation. In this case, the task is to decide which of the two alternatives is more likely to result from the situation described in the sentence. We call this second category the *result category*. Even though the category does not influence the way the background knowledge is selected, it is necessary to use different approaches for the two categories when combining this background knowledge with automated reasoning methods.

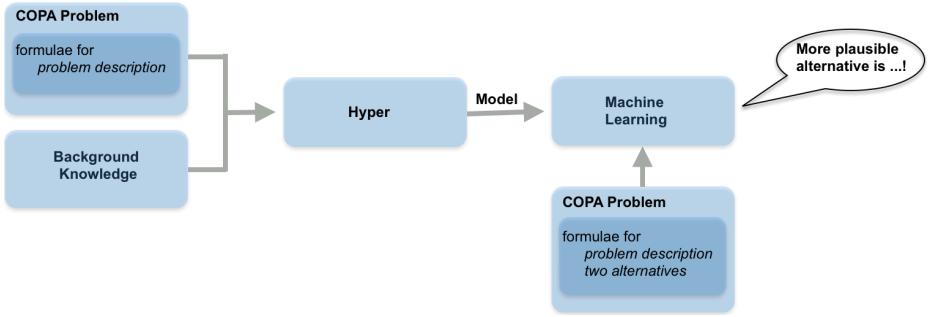

Fig. 4: Addressing a problem in the *result* category: Using the selected background knowledge together with Hyper and machine learning.

Figure 4 depicts how to tackle a problem from the result category in order to determine the more plausible alternative result. Please note that the selected background knowledge does not only consist of axioms stemming from the knowledge base used as a source for background knowledge but also contains the connecting formulae which were created as depicted in Figure 3. First, this background knowledge is combined with the logical formulae representing the description of the benchmark problem. The resulting set of formulae serves as input for a theorem prover, in our case the Hyper prover. Hyper constructs a model for the set of formulae which, together with the logical representation of the two alternatives, is used by machine learning techniques to determine which of the two alternatives is more plausible. In the remainder of this paper, we focus on problems belonging to the result category.

3.3 Lessons Learnt so far

We created a prototypical implementation of the workflow depicted in the previous section. Our implementation is able to take a problem from the COPA or Triangle-COPA challenge, it selects appropriate background knowledge, generates a connecting set of formulae and feeds everything into Hyper. The machine learning component inspecting the generated model is in an experimental phase and is addressed in Section 3.4 below.

Issues with Inconsistencies

COPA challenge We performed a very preliminary experiment to test this workflow. From the COPA benchmark set we selected 100 problems. Feeding these examples into the workflow resulted finally in 100 proof tasks for Hyper and we learned a lot — about problems which have to be solved. Hyper found 37 proofs and 57 models; the rest are time-outs. One problem we encountered is that some contradictions leading to a proof are introduced by selecting too general hypernyms from WordNet. E.g. the problem description of example 1 given

in Figure 1 is transformed into the following first-order logic formula by Boxer:

$$\exists A(ngrassC(A) \wedge \exists B, C, D(rover(B, A) \wedge rpatient(B, C) \wedge ragent(B, D) \\ \wedge vcast(B) \wedge nshadow(C) \wedge nbody(D) \wedge rof(D, C) \wedge nperson(C)))).$$

From WordNet the system extracted the information, that 'individual' is a hypernym of 'shadow' and 'collection' is a hypernym of 'person' leading to the two connecting formulae:

$$\forall X(nshadow(X) \rightarrow individual(X)) \\ \forall X(nperson(X) \rightarrow collection(X)).$$

The selection from OpenCyc resulted among others in the axiom

$$\forall X \neg(collection(X) \wedge individual(X)).$$

These formulae together lead to a closed tableau —a proof of unsatisfiability — which has nothing to do with one of the alternatives that the sun was rising or the grass was cut.

To remedy this problem, we use a tool called KNEWS[4] [1] to disambiguate Boxer's output. This tool calls the Babelfy[5] [21] service to link entities to BabelNet[6] [23]. Babelfy is a multilingual, graph-based approach to entity linking and word sense disambiguation. BabelNet is a multilingual encyclopedic dictionary and a semantic network. Since the BabelNet entries are linked to Wordnet synsets, this tool provides the suitable Wordnet synset for predicate names generated by Boxer. In a second run of the experiment we only used the disambiguated results to construct a bridging set of formulae and to select background knowledge. It turned out that the selected background knowledge is much more focused on the problem under consideration. Furthermore, only one of the 100 COPA problems we tested, was inconsistent. So we solved this first problem by disambiguating Boxer's output.

The one contradiction which still occurred in the second experiment stems directly from inconsistencies in the knowledge base used as source for background knowledge (in our case OpenCyc). E.g. the two formulae

$$\forall X\, speed(fqpquantityfnspeed(X)) \\ \forall X \neg speed(X)$$

were selected immediately leading to a contradiction which again does not have to do anything with the two alternatives about the sun rising or the grass being cut. This illustrates that we have to find a way to deal with inconsistent background knowledge.

[4] Many thanks to Valerio Basile for being so kind to share KNEWS. (Available at: https://github.com/valeriobasile/learningbyreading)
[5] Available at: http://babelfy.org
[6] Available at: http://babelnet.org

Triangle-COPA challenge We used the workflow depicted in Figure 3 for the 100 problems given in the Triangle-COPA challenge with a timeout set to 1000 sec. Due to the structure of the problems, we treated all problems in the Triangle-COPA challenge as problems belonging to the afore-described result category. In our first experiments, Hyper constructed 12 models, 65 proofs and 17 timeouts. The remaining 6 problems caused an error. Inspection of the 65 proofs revealed that 38 were caused by the following formula selected from Cyc:

$$\forall X \neg action(X) \tag{1}$$

As soon as $action(i)$ can be derived for an individual i, this formula leads to a contradiction. Since the topic of all Triangle-COPA problems are interpersonal relationships, it is reasonable that instances of the *action* predicate are derived in many examples. Formula (1) itself does not provide interesting information for our scenario which is why we removed it from our version of OpenCyc.

We restarted the workflow for the modified background knowledge. This resulted in 15 models, 42 timeouts, 37 proofs and 6 errors. It is remarkable that the major part of the 22 examples which were unsatisfiable due to Formula (1) in the first experiment, led to timeouts in our second experiment. Only 3 of these problems led to models. We are planning to further improve these results by adding new sources for background knowledge as described in the next section.

Insufficient Background Knowledge

COPA challenge Another challenge when combining problems with background knowledge is the lack of appropriate background knowledge. Consider the example number 13 from the COPA challenge which we presented in Figure 1. The background knowledge selected for this example contains formulae on iceskating:

$$\forall X(iceskate(X) \rightarrow isa(X, c\_iceskate)).$$

However, the information that freezing of a pond results in a surface suitable for skating, is missing. This explains, why not enough inferences were performed and the constructed model does not contain information on ice skating.

One explanation for the lack of inferences is the fact that we are currently only using OpenCyc as a source of background knowledge. We are planning to remedy this situation by including different other sources of background knowledge like ConceptNet [18], the Suggested Upper Merged Ontology (SUMO) [24, 25], the Human Emotion Ontology (HEO) [12] and the Emotion Ontology (EMO) [14].

ConceptNet is a semantic network containing large amounts of commonsense knowledge. This graph consists of labeled nodes and edges. The nodes are also called concepts and represent words or word senses. The edges are relations between these concepts and represent common-sense knowledge that connect the concepts. Relating to the COPA problem described afore, ConceptNet contains very helpful knowledge like the fact that in winter one likes to skate.

$$winter\text{–}CausesDesire \rightarrow skate$$

SUMO is a very large upper ontology containing knowledge which could be helpful as background knowledge. For example, SUMO contains the knowledge that *icing* is a subclass of *freezing* which could be helpful for our benchmark problem. One very interesting point is that there is a mapping from SUMO to WordNet synsets. This will be very helpful during the creation of formulae bridging from the vocabulary of the benchmark problem and the synonyms and hypernyms to the vocabulary used in SUMO.

Triangle-COPA challenge When testing the workflow depicted in Figure 3, experiments produced 15 models. Please note that, since our background knowledge does not contain eventualities, we remove the eventuality from the formulae describing the problem, meaning that we map $shake(e2, lt)$ to $shake(lt)$. Closer inspection of these models revealed that some helpful inferences were made. Considering the model constructed for example problem no. 44 given in Figure 2 shows that the following ground instances were derived: $leave(lt)$, $move(lt)$, $tremble(lt)$, $judder(lt)$ and $shiver(lt)$. While these inferred ground instances look encouraging, the final step of deducing $cold(lt)$ is still missing. This situation could be remedied by the use of ConceptNet, which contains fitting additional knowledge "You would shiver because it was cold', represented as:

$$shiver\text{–}MotivatedByGoal \rightarrow it\ be\ cold$$

This is why we are planning to integrate ConceptNet as background knowledge. Another interesting source for additional background knowledge are the HEO and EMO ontology. Both these ontologies contain information on human emotions which is very suitable for the Triangle-COPA challenge since its problems consist of descriptions of small episodes on interpersonal relationships.

3.4 Ranking of Proofs and Proof Attempts

When using the automated reasoning system Hyper within the LogAnswer system we already had to tackle the problem that the prover nearly never found a complete proof of the given problem. In order to find a best answer of the system we had to compare several proofs, or rather proof attempts. For this ranking we used machine learning to find the best proof rsp. answer. We are planning to use a similar approach for the commonsense benchmarks.

In the sequel we describe how to use machine learning techniques for problems of the result category and present first experimental results. In the workflow depicted in Figure 4, we construct a tableau for $P \cup BG$, where P is the problem description and BG is the background knowledge. This tableau may contain open and closed branches. The closed branches are parts of a proof and the open branches either represent a model (and hence no closed tableau exists) or they are only open because of a time-out for this branch. With the help of this tableau, we try to decide which of the two alternatives $E1$ and $E2$ is 'closer' to a logical consequence of $P \cup BG$.

In the LogAnswer system we gave the two answers to humans to decide which is closer to a logical consequence and we then used this information to train a machine learning system. For the scenario of the COPA and Triangle-COPA benchmarks we designed a preliminary study, which aims at using the information about the tableau created for $P \cup BG$ together with information from formulae of the problem and the background $P \cup BG$ to generate examples for training.

We restricted our preliminary study to propositional logic and analyzed tableaux created by the Hyper prover for randomly created sets of clauses. For each pair of propositional logic variables p and q occurring in a clause set, we were interested in the question if p or q is 'closer' to a logical consequence. We reduced this question to a classification problem: for each pair of variables p and q, the task is to learn if $p < q$, $p > q$ or $p = q$, where $p < q$ means, that q is 'closer' to a logical consequence than p and $p = q$ means that p's and q's 'closeness' to a logical consequence is equal. Consider the following set of clauses:

$$\begin{aligned} p0 \\ p4 &\rightarrow p2 \vee p3 \vee p7 \\ p0 &\rightarrow p4 \\ p3 \wedge p5 &\rightarrow p6 \\ p3 \wedge p5 \wedge p8 &\rightarrow p1 \\ p2 &\rightarrow \bot \end{aligned}$$

Clearly, $p0$ and $p4$ are logical consequences of this clause set. Therefore $p0 = p4$ and $p0 > q$ for all other variables q. On the other hand, from $p2$ it is possible to deduce a contradiction, which leads to $p2 < q$ for all other variables q. Comparing $p6$ and $p1$ is a little bit more complicated. Neither of these variables is a logical consequence. However, assuming $p3$ and $p5$ to be true, allows to deduce $p6$ but not $p1$. In oder to deduce $p1$ it is necessary to assume not only $p3$ and $p5$ to be true but also $p8$. Therefore $p1 < p6$.

To use machine learning techniques to classify this kind of examples, we represent each pair of variables (p, q) as an instance of the training examples and we provide the information, which of the three relations $<, >, =$ is correct for p and q. Each of these instances contains 22 attributes. Some of these attributes represent information on the clause set like the proportion of clauses with p or q in the head as well as rudimentary dependencies between the variables in the clause set. In addition to that, we determine attributes representing information on the hypertableau for the set of clauses like the number of occurrences of p and q in open branches. Furthermore, we determine an attribute mimicking some aspects of abduction by estimating the number of variables which have to be assumed to be true in order to deduce p or q respectively. This allows us to perform comparisons like the one between $p1$ and $p6$ in the above example. Of course, we also take into account whether one of the two variables is indeed a logical consequence.

	< (predicted)	> (predicted)	= (predicted)
< (actual)	5,595	78	33
> (actual)	90	5,589	27
= (actual)	9	5	772

Table 1: Confusion matrix for classifying the test set with the learnt decision tree. The numbers occurring in the diagonal represent all correctly classified instances, whereas the other cells list incorrectly classified instances.

For the first experiments, 1,000 sets of clauses each consisting of about 10 clauses and containing about 12 variables were randomly generated and used to create a training set. For each pair of variables occurring in one of the clause sets, an instance was generated. All in all this led to 123,246 examples for training purposes. In these examples, the classes $<$ and $>$ each consists of 57,983 examples and the class $=$ of 7,280 examples. We used the J48 tree classifier implemented in the Weka [13] system to construct a decision tree for the training set. This classifier implements the C4.5 algorithm. We tested the generated decision tree with a test set which was generated from 100 randomly generated sets of clauses different from the clause sets used for the training examples. This resulted in a test set consisting of 12,198 instances. The learnt decision tree correctly classified 98.02 % instances of our test set. Table 1 provides information on correctly and incorrectly classified instances of the different classes.

We are aware that automatically classifying the test set might introduce errors into the test set and therefore tampers the results. Since it is very labor-intensive to manually generate test data, we only created test instances from two clause sets manually. For this much smaller test set the generated decision tree was able to correctly classify 80 % of the instances.

In the next step, we are planning to expand our experiments to clause sets given in first-order logic. When creating the instances of the training examples for first-order logic, attributes different from the ones in the previous experiment have to be considered, since unification has to be taken into account.

4 Normative Statementes

When considering commonsense reasoning problems, normative statements can be very useful as well. In this section we discuss the presentation of normative rules in commonsense reasoning and in multi-agent systems.

Norms in Commonsense Reasoning In [8] we introduced some ideas how to add normative statements to the background knowledge used to tackle the Triangle-COPA challenge. The main idea of these normative statements was to express information about met and unmet expectations. To formalize these

statements, standard deontic logic was used. Standard deontic logic (SDL) corresponds to the well-known modal logic K together with a seriality axiom D : $\Box P \rightarrow \Diamond P$. In this logic, the $\Box$-operator is interpreted as 'it is obligatory that' and the $\Diamond$ as 'it is permitted that'. The $\Diamond$-operator can be defined by $\Diamond P \equiv \neg\Box\neg P$. The seriality axiom in SDL states that, if a formula has to hold in all reachable worlds, then there exists such a world. With the deontic reading of $\Box$ and $\Diamond$ this means: Whenever the formula P ought to be, then there exists a world where it holds. In consequence, there is always a world, which is ideal in the sense that all the norms formulated by 'the ought to be'-operator hold.

Considering the Triangle-COPA challenge, it sounds reasonable to formalize the fact that one should defend friends if they are under attack. This can be accomplished by a set of deontic logic formulae which are the set of ground instances of the following statement:

$$\mathit{friend}(X, Y) \wedge \mathit{attack}(Z, X) \rightarrow \Box \mathit{defend}(Y, X). \tag{2}$$

Since formula (2) contains variables, it is not a SDL formula. However, we use it as an abbreviation for its set of ground instances. In the same way we could formalize that a person is disappointed if she is under attack and a friend does not hurry to her defense.

Deontic logic is not only the logic of choice when formalizing knowledge about norms in interpersonal relationships but also for the formalization of ethical codes for agents.

Norms in Robot Ethics In multi-agent systems, there is a challenging area of research, namely the formalization of 'robot ethics'. It aims at defining formal rules for the behavior of agents and to prove certain properties. As an example consider Asimov's laws, which aim at regulating the relation between robots and humans. In [3], the authors depict a small example of two surgery robots obeying ethical codes concerning their work. These codes are expressed by means of MADL, which is an extension of standard deontic logic with two operators. In [22], an axiomatization of MADL is given. Further, it is asserted, that MADL is not essentially different from standard deontic logic. This is why we use SDL to model the example.

In the example, there are two robots *ag1* and *ag2* in a hospital. For the sake of simplicity, each robot can perform one specific action: *ag1* can terminate a person's life support and *ag2* can delay the delivery of pain medication. In [3], four different ethical codes J, $J^\star$, O and $O^\star$ are considered:

- "If ethical code J holds, then robot *ag1* ought to take care that life support is terminated." This is formalized as:

$$J \rightarrow \Box \mathit{act}(\mathit{ag1}, \mathit{term}) \tag{3}$$

- "If ethical code $J^\star$ holds, then code J holds, and robot *ag2* ought to take care that the delivery of pain medication is delayed." This is formalized as:

$$J^\star \rightarrow J \wedge J^\star \rightarrow \Box \mathit{act}(\mathit{ag2}, \mathit{delay}) \tag{4}$$

– "If ethical code O holds, then robot *ag2* ought to take care that delivery of pain medication is not delayed." This is formalized as:

$$O \rightarrow \Box\neg act(ag2, delay) \tag{5}$$

– "If ethical code $O^\star$ holds, then code O holds, and robot *ag1* ought to take care that life support is not terminated." This is formalized as:

$$O^\star \rightarrow O \wedge O^\star \rightarrow \Box\neg act(ag1, term) \tag{6}$$

Further we give a slightly modified version of the evaluation of the acts of the robots, as stated in [3], where $(+!!)$ denotes the most and $(-!!)$ the least desired outcome. Note that terms like $(+!!)$ are just propositional atomic formulae here.

$$act(ag1, term) \wedge \quad act(ag2, delay) \rightarrow (-!!) \tag{7}$$
$$act(ag1, term) \wedge \neg act(ag2, delay) \rightarrow (-!) \tag{8}$$
$$\neg act(ag1, term) \wedge \quad act(ag2, delay) \rightarrow (-) \tag{9}$$
$$\neg act(ag1, term) \wedge \neg act(ag2, delay) \rightarrow (+!!) \tag{10}$$

These formulae evaluate the outcome of the robots' actions. It makes sense to assume, that this evaluation is effective in all reachable worlds. This is why we add formulae stating that formulae (7)–(10) hold in all reachable worlds. For example, for (7) we add:

$$\Box(act(ag1, term) \wedge act(ag2, delay) \rightarrow (-!!)) \tag{11}$$

Since our example does not include nested modal operators, the formulae of the form (11) are sufficient to spread the evaluation formulae to all reachable worlds. The normative system $\mathcal{N}$ formalizing this example consists of the formalization of the four ethical codes and the formulae for the evaluation of the robots actions.

A possible query would be to ask if the most desirable outcome $(+!!)$ will come to pass if ethical code $O^\star$ is operative. This query can be translated into a satisfiability test. If $\mathcal{N} \wedge O^\star \wedge \Diamond\neg(+!!)$ is unsatisfiable, then ethical code $O^\star$ ensures outcome $(+!!)$.

Since Hyper is able to decide the description logic $\mathcal{SHIQ}$ and standard deontic logic formulae can be translated into description logic knowledge bases, we can use Hyper for this satisfiability test. We obtain the desired result namely that (only) ethical code $O^\star$ leads to the most desirable behavior $(+!!)$.

5 Conclusion

We presented an approach to tackle benchmarks for commonsense reasoning. This approach relies on large existing ontologies as a source for background knowledge and combines different techniques like theorem proving and machine learning with tools for natural language processing. With the help of a prototypical implementation of our approach, we conducted some experiments with

problems from the COPA challenge. We presented our experiences made in these experiments together with possible solutions for the problems occurring in the examples considered.

Future work aims at the integration of additional sources of background knowledge as well as improving the bridging between the vocabulary used in the benchmarks and the background knowledge.

References

1. V. Basile, E. Cabrio, and F. Gandon. Building a general knowledge base of physical objects for robots. In *The Semantic Web. Latest Advances and New Domains*, 2016.
2. M. Bender, B. Pelzer, and C. Schon. System description: E-KRHyper 1.4 – extensions for unique names and description logic. In M. P. Bonacina, editor, *CADE-24*, LNCS 7898, pages 126–134. Springer, 2013.
3. S. Bringsjord, K. Arkoudas, and P. Bello. Toward a general logicist methodology for engineering ethically correct robots. *IEEE Intelligent Systems*, 21(4):38–44, 2006.
4. J. R. Curran, S. Clark, and J. Bos. Linguistically motivated large-scale NLP with C&C and Boxer. In *Proceedings of the ACL 2007 Demo and Poster Sessions*, pages 33–36, Prague, Czech Republic, 2007.
5. D. A. Ferrucci. IBM's Watson/DeepQA. *SIGARCH Comput. Archit. News*, 39(3):– , June 2011.
6. U. Furbach, I. Glöckner, H. Helbig, and B. Pelzer. Logic-based question answering. *KI - Künstliche Intelligenz*, 2010. Special Issue on Automated Deduction, Febr. 2010.
7. U. Furbach, A. Gordon, and C. Schon. Tackling benchmark problems for commonsense reasoning. In *Proceedings of Bridging - Workshop on Bridging the Gap between Human and Automated Reasoning*, 2015.
8. U. Furbach, B. Pelzer, and C. Schon. Automated reasoning in the wild. In *CADE-25 – The 25th International Conference on Automated Deduction*, volume 9195 of *Lecture Notes in Artificial Intelligence*. Springer, 2015.
9. U. Furbach and C. Schon. Commonsense reasoning meets theorem proving. In *Proceedings of the 1st Conference on Artificial Intelligence and Theorem Proving AITP'16, Obergurgl, Austria*, 2016.
10. U. Furbach and C. Schon. Commonsense reasoning meets theorem proving. In *to appear in Proceedings of Bridging-20016 - Workshop on Bridging the Gap between Human and Automated Reasoning*, 2016.
11. A. S. Gordon. Commonsense interpretation of triangle behavior. In D. Schuurmans and M. P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3719–3725. AAAI Press, 2016.
12. M. Grassi. *Biometric ID Management and Multimodal Communication: Joint COST 2101 and 2102 International Conference, BioID_MultiComm 2009, Madrid, Spain, September 16-18, 2009. Proceedings*, chapter Developing HEO Human Emotions Ontology, pages 244–251. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
13. M. A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.

14. J. Hastings, W. Ceusters, B. Smith, and K. Mulligan. *Modeling and Using Context: 7th International and Interdisciplinary Conference, CONTEXT 2011, Karlsruhe, Germany, September 26-30, 2011. Proceedings*, chapter The Emotion Ontology: Enabling Interdisciplinary Research in the Affective Sciences, pages 119–123. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.
15. C. Kaliszyk, S. Schulz, J. Urban, and J. Vyskocil. System description: E.T. 0.1. In A. P. Felty and A. Middeldorp, editors, *Proceedings of CADE-25, Berlin, Germany, 2015*, volume 9195 of *LNCS*. Springer, 2015.
16. D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
17. H. J. Levesque. The Winograd Schema Challenge. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI, 2011.
18. H. Liu and P. Singh. ConceptNet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, Oct. 2004.
19. N. Maslan, M. Roemmele, and A. S. Gordon. One hundred challenge problems for logical formalizations of commonsense psychology. In *Twelfth International Symposium on Logical Formalizations of Commonsense Reasoning, Stanford, CA*, 2015.
20. G. A. Miller. WordNet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
21. A. Moro, A. Raganato, and R. Navigli. Entity Linking meets Word Sense Disambiguation: a Unified Approach. *Transactions of the Association for Computational Linguistics (TACL)*, 2:231–244, 2014.
22. Y. Murakami. Utilitarian deontic logic. In *in 'Proceedings of the Fifth International Conference on Advances in Modal Logic (AiML 2004*, pages 288–302, 2004.
23. R. Navigli and S. P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.
24. I. Niles and A. Pease. Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM, 2001.
25. A. Pease. *Ontology: A Practical Guide*. Articulate Software Press, Angwin, CA, 2011.
26. M. Roemmele, C. A. Bejan, and A. S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, 2011.
27. F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from Wikipedia and WordNet. *Web Semant.*, 6(3):203–217, Sept. 2008.