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 the 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. We furthermore report about a preliminary experiment for finding most plausible results.

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. In this paper, we describe the progress we made so far in creating a system able to tackle commonsense reasoning benchmarks. The system uses techniques from automated theorem proving but also relies on natural language processing, large ontologies as background knowledge and machine learning.

In Section 2 several benchmarks for the area of commonsense reasoning are introduced. Section 3 describes which techniques will be combined in order to create a system able to perform commonsense reasoning. Challenges that have to be faced when including large ontologies as background knowledge are described in Section 4. In order to find the more plausible answer to the benchmark problems, machine learning techniques come to use. Section 5 presents first experimental results from this area.

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

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.

2 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 [13] 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[1] [21] 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.

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[2] [15]. 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. Approaches to tackle the Triangle-COPA challenge can be found in [15], [5] and [7].

In this paper we focus on the COPA challenge. However most parts of our approach can be used for the problems of the Triangle-COPA challenge as well.

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

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

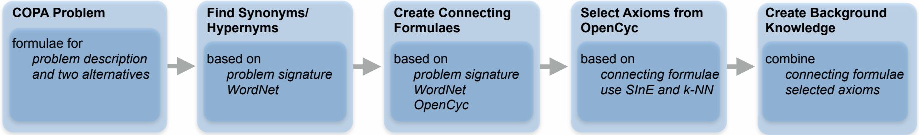

Fig. 2: Gathering background knowledge for a benchmark problem. The starting formulae are generated by transforming the natural language problems of COPA into first-order logic using the Boxer system.

3 Combination of Techniques

As described afore, the creation of a system for commonsense reasoning requires the combination of techniques from different areas [6].

Even gathering appropriate background knowledge for a specific benchmark problems requires the use of different techniques. Figure 2 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.
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 [3] 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 2 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 [16] 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 [12], 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 [11] 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

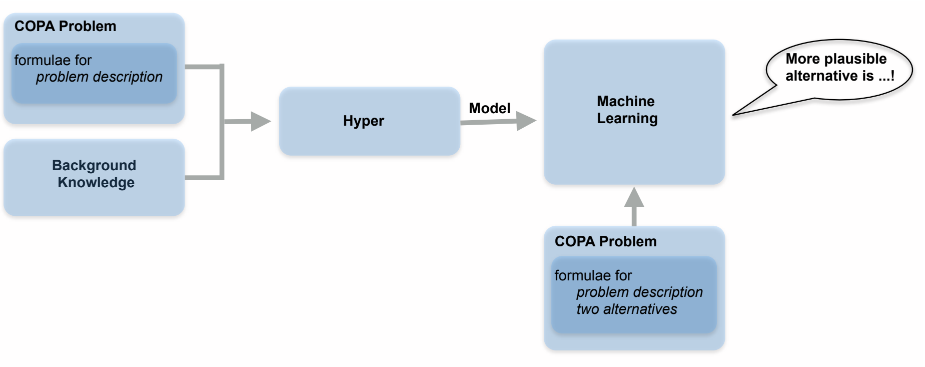

Fig. 3: Addressing a problem in the *result* category: Using the selected background knowledge together with Hyper and machine learning.

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 thought 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.

Figure 3 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 2. 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 [2]. 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.

When dealing with problems in the cause category, we have to solve two reasoning problems. Figure 4 depicts the workflow for this category. In the first step, the background knowledge is combined with the logical formulae representing the first alternative. The resulting set of formulae serves as an input for the Hyper theorem prover, which constructs a model M_1. In the second step, the background knowledge is combined with the logical formulae representing the second alternative and the resulting set is passed to Hyper which constructs a model M_2. Then both models are inspected in order to determine which of the

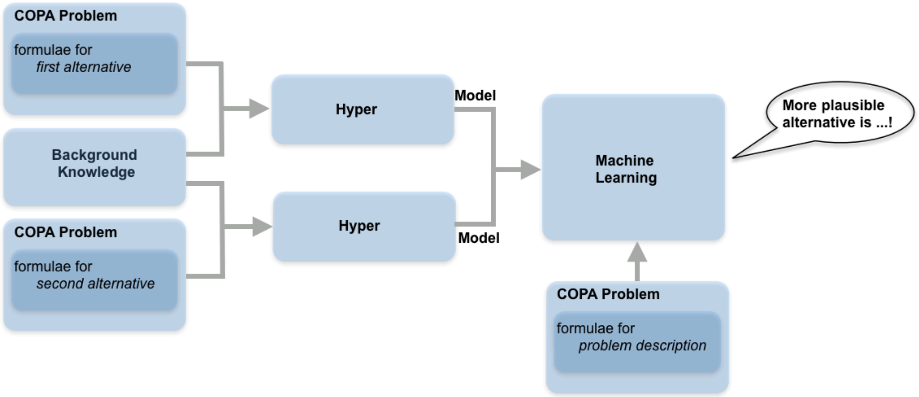

Fig. 4: Addressing a problem in the *cause* category: Using the selected background knowledge to call Hyper for both alternatives separately and use machine learning to choose the more plausible alternative.

two models is closer to the formulae representing the description of the problem. This inspection of the models is still future work. We are planning to accomplish this with the help of machine learning.

4 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 challenge, 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 5 below.

4.1 Issues with Inconsistencies

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 prove 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:

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

The selection from OpenCyc resulted among others in the axiom

$$\forall X \neg(\mathit{collection}(X) \land \mathit{individual}(X)).$$

These formulae together lead to a closed tableau —a proof of unsatisfiability— because of a contradiction between a shadow which stems from a person, whereas WordNet gives that a shadow is an individual, a person is a collection and together with the Cyc axiom we get the contradiction. This 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 [1] [3] to disambiguate Boxer’s output. This tool calls the Babelfy[4] [17] service to link entities to BabelNet[5] [18]. 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 experiments stems directly from inconsistencies in the knowledge base used as source for background knowledge (in our case OpenCyc). E.g. the two formulae

$$\forall X\, \mathit{speed}(\mathit{fqpquantityfnspeed}(X))$$
$$\forall X \neg \mathit{speed}(X)$$

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.

4.2 Insufficient Background Knowledge

Another challenge when combining problems with background knowledge is the lack of appropriate background knowledge. Consider the example number 13

[3] Many thanks to Valerio Basile for being so kind to share his tool for disambiguation. The tool is available at https://github.com/valeriobasile/learningbyreading

[4] Available at: http://babelfy.org

[5] Available at: http://babelnet.org

from the COPA challenge which we presented in Figure 1. The background knowledge selected for this example contains formulae on skating like for example:

$$\forall X(\mathit{skateboard}(X) \rightarrow \mathit{device\_usercontrolled}(X))$$

and even on iceskating

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

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

In general there are two possible explanations for the lack of inferences:

1. The way the bridging between the vocabulary used in the benchmark problem and the vocabulary used in OpenCyc is still very prototypical. We heavily rely on Boxer's output to construct the connecting formulae. However sometimes Boxer marks adjectives as nouns and vice versa which misguides the search for synonyms and hypernyms in WordNet. We are planning to improve this by updating to the most recent Boxer version.
2. Currently, we are only using OpenCyc as a source of background knowledge. However this is not sufficient. We are planning to remedy this situation by including different other sources of background knowledge like ConceptNet [14], the Suggested Upper Merged Ontology (SUMO) [19, 20], the Human Emotion Ontology (HEO) [8] and the Emotion Ontology (EMO) [10].
   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.

   $$\mathit{winter}{-}\mathit{CausesDesire} \rightarrow \mathit{skate}$$

   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.
   Due the fact that the structure of problems of the Triangle-COPA challenge is very similar to the structure of the problems of the COPA challenge, our approach can be used for the Triangle-COPA challenge without much changes. Since the problems given in the Triangle-COPA challenge consist of descriptions of small episodes on interpersonal relationships, the HEO and EMO ontology, both containing information on human emotions, provide useful background knowledge.

5 Ranking of Proofs and Proof Attempts

When using the automated reasoning system Hyper within the Deep Question Answering system LogAnswer ([4]) we already had to tackle the problem that the prover nearly never found a complete proof of the given problem. Instead we had to delete some subgoals of the proof-task which could not be solved within a given time-bound — we called this relaxation. In order to find a best answer of the system we had to compare several proofs, or rather proof attempts, because of the relaxations. For this ranking we used machine learning to find the best proof rsp. answer. We are planning to use a similar approach for the afore-mentioned benchmarks.

Like depicted in Figure 4, for the cause category two models are generated. Each of these models corresponds to knowledge which was derived using one of the two alternatives. In future work we are planning to use these two models together with the formulae representing the description of the situation in order to find out which alternative is the more plausible one. For this we are planning to use machine learning techniques to determine if the formulae describing the situation is rather a logical consequence of the first or the second model. Up till now, we focused our research on the result category which is why we cannot provide further results for the cause category.

In the sequel we describe how to use machine learning techniques for problems of the result category and give results of a first experiment. The situation is such that we have a problem description P, in our COPA example above the logical representation of 'The pond froze over for the winter.' together with two possible answers $E1$ and $E2$.

In the workflow described until now, we constructed a tableau for $P \cup BG$, where BG is the background knowledge as discussed in Section 3. This tableau may contain open and closed branches. The closed branches are parts of a proof and the open branches either are 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 other words we try to decide if the constructed tableau can be rather closed by adding alternative $E1$ or by adding alternative $E2$.

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 these kind of examples, we represent each pair of variables (p, q) as an instance of the trainings 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.

For the first experiments, 1,000 sets of clauses each consisting of about 10 clauses and containing about 12 variables were randomly generated. These sets of clauses were analyzed 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 [9] system to construct a decision tree for the training set. This classifier implements the C4.5 algorithm. The result was a decision tree for our training examples. We tested this 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

	< (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.

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, depending on the classifier used, we reached percentages of correctly classified instances of up to 80 %.

In the next step, we are planning to expand our experiments to clause sets $P \cup BG$ and alternatives $E1$ and $E2$ given in first-order logic. When creating the instances of the trainings examples for first-order logic, attributes different from the ones in the previous experiment have to be considered. For example it is not sufficient to have an attribute indicating the proportion of open branches containing a certain subgoal of $E1$. Since both the open branch and the subgoal of $E1$ are given in first-order logic, unification has to be taken into account. In this case the proportion of open branches containing an atom which can be unified with one of the subgoals of $E1$ constitutes an interesting attribute. Another interesting attribute would take relaxation into account: for an open branch not containing any atom which can be unified with one of the subgoals of $E1$, it is interesting if this branch contains an atom which can be unified with a generalization of one of the subgoals in $E1$. This attribute allows us to mimic the relaxation like it is used in the LogAnswer system.

6 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. 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.
4. U. Furbach, I. Glöckner, and B. Pelzer. An application of automated reasoning in natural language question answering. *AI Commun.*, 23(2-3):241–265, 2010.
5. 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.
6. 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.
7. 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.
8. 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.
9. 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.
10. 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.
11. 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.
12. D. B. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.
13. 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.
14. H. Liu and P. Singh. ConceptNet — a practical commonsense reasoning tool-kit. *BT Technology Journal*, 22(4):211–226, Oct. 2004.

15. 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.
16. G. A. Miller. WordNet: a lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
17. 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.
18. 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.
19. 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.
20. A. Pease. *Ontology: A Practical Guide.* Articulate Software Press, Angwin, CA, 2011.
21. 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.