Künstliche Intelligenz 33:293–299 (2019)
Final publication available at https://link.springer.com/article/10.1007/s13218-019-00601-5.

The CoRg Project – Cognitive Reasoning

Claudia Schon · **Sophie Siebert** · **Frieder Stolzenburg**

Received: 18 January 2019 / Accepted: 18 January 2019

Abstract The term cognitive computing refers to new hardware and/or software that mimics the functioning of the human brain. In the context of question answering and commonsense reasoning this means that the reasoning process of humans shall be modeled by adequate technical means. However, since humans do not follow the rules of classical logic, a system designed to model these abilities must be very versatile. The aim of the CoRg project (Cognitive Reasoning) is to successfully complete a reasoning task with commonsense reasoning. We address different benchmarks with focus on the COPA benchmark set (Choice of Plausible Alternatives). Since humans naturally use background knowledge, we have to deal with large background knowledge bases and must be able to reason with multiple input formats and sources in the CoRg system, in order to draw explainable conclusions. For this, we have to find appropriate logics for cognitive reasoning. For a successful reasoning system, nowadays it seems to be important to combine automated reasoning with machine learning technology like recurrent neural networks.

Keywords cognitive reasoning · commonsense reasoning · automated reasoning · machine learning

1 Introduction

Cognitive reasoning is a sub-discipline of artificial intelligence that attempts to model the way people draw conclusions in everyday situations. However, since humans do not follow the rules of classical logic during commonsense reasoning, a system designed to model these abilities must be very versatile. Humans are able to draw meaningful conclusions despite incomplete, possibly even inconsistent knowledge. They can deal with norms and conflicting norms and are able to rethink their conclusions as new information arrives. The CoRg project has the ambitious goal to develop a system for cognitive reasoning. The main challenges that the project addresses are:

1. **Gathering and using background knowledge:** Cognitive reasoning requires enormous amounts of background knowledge describing everyday experience humans use for reasoning. This background knowledge has to be constructed by combining appropriate sources. Furthermore, the cognitive system must contain mechanisms to deal with the sheer size of this knowledge.
2. **Reasoning with multiple formats:** Logical reasoning alone is not sufficient to model human reasoning. A cognitive system has to be able to handle natural language and deliberate about different conclusions which may be conflicting. We achieve this by integrating machine learning algorithms.
3. **Finding appropriate logics for cognitive reasoning:** Logics used for cognitive reasoning have to capture the versatility of human reasoning. A system for cognitive reasoning should therefore include a combination of different techniques and logics.
4. **Drawing explainable conclusions:** It is not sufficient for a system to provide only a yes or no answer. In order to be comprehensible, the responses of a system must be accompanied by a justification.

In order to evaluate the CoRg system, we use benchmarks from commonsense reasoning. Numerous benchmarks have been presented in recent years that are suitable for testing a system for cognitive reasoning. The examples range from benchmarks that require causal reasoning in everyday situations, such as the COPA Challenge (Choice of Plausible Alternatives) [22], to benchmarks that address difficult cases of pronoun disambiguation, such as the Wino-

The authors gratefully acknowledge the support of the German Research Foundation (DFG) under the grants SCHO 1789/1-1 and STO 421/8-1 *CoRg* – Cognitive Reasoning.

```
1: My body cast a shadow over the grass.
What was the CAUSE of this?
A1: The sun was rising.
A2: The grass was cut.

18: It got dark outside.
What happened as a RESULT?
A1: Snowflakes began to fall from the sky.
A2: The moon became visible in the sky.
```

Figure 1 Problem 1 and 18 from the COPA benchmark set.

grad Schema Challenge [14], to benchmarks that focus on human relationships and emotions, such as the Triangle-COPA Challenge [16]. Although we initially concentrate on the COPA Challenge in the CoRg project, the same approach can also be applied to other commonsense reasoning benchmarks. However, it is expected that problems like the Winograd Schema Challenge, where linguistic knowledge plays an important role, will produce less good results.

The rest of this article is structured as follows: Sect. 2 introduces the structure of the cognitive computing system currently being developed in the CoRg project. Sect. 3 to 6 provide details on the four challenges mentioned above and present the solutions sought in the CoRg project. In Sect. 7 we present what we have learned so far from tackling the challenges, before we end up with conclusions in Sect. 8.

2 The Structure of the CoRg System

The objective of CoRg is to develop a system for cognitive computing. It is an extension of the system described in [7] and will be evaluated using benchmarks from the commonsense reasoning area. While we aim at being able to address different benchmarks, we currently focus on the COPA benchmark set (Choice of Plausible Alternatives) [22] consisting of 1,000 problems. We use the 500 development examples as training and validation set and the 500 test examples as test set. The COPA problems have a common scheme, consisting of a sentence, describing a situation, together with a question and two possible answers to choose from (cf. Fig. 1). They are split evenly into the categories cause and result, indicating the relationship between the question and the answer. Forward causal reasoning is necessary for the problems in the result category (problem 18 in Fig. 1), backward causal reasoning for the problems of the cause category (problem 1 in Fig. 1).

We now describe the structure of the CoRg system exemplarily by means of the processing of a task from the COPA problem set. Given the task, a variety of steps is performed, as depicted in Fig. 2. The first step within the CoRg system is to transform each natural language sentence, i.e., the problem description and both answers, into a first-order logic formula using KNEWS [2], a tool that performs semantic parsing including some form of pronoun disambiguation, word sense disambiguation, and entity linking. Predicate symbols used in the formulae created by KNEWS correspond to words (e.g. nouns, verbs and adjectives) of the original text. Besides this transformation, KNEWS together with WordNet [17] also performs word-sense disambiguation and provides a so-called *synset* ID, that is a grouped set of cognitive synonyms for every meaningful word (e.g. nouns and verbs) occurring in the COPA problem. For each synset, both hyponyms and hypernyms, also taken from WordNet, are determined. For the first-order logic formula produced by KNEWS and the determined hyponyms and hypernyms, the CoRg system searches for relevant background knowledge within large knowledge bases like SUMO [19], Adimen-SUMO [1], ResearchCyc [13], YAGO [26] or ConceptNet [24].

The gathered information together with the logical representation of the original text is then fed into Hyper. Hyper [3] is an automated theorem prover based on the E-hyper tableau calculus. Hyper receives the information for each question-answering task at once, not sequentially as the text. Because of the built-in first-order logic calculus, it performs monotonic reasoning. Yet it can deal with incomplete knowledge by constructing models which are processed further in the subsequent steps. So, for satisfiable problems, Hyper is able to construct models. In the case of a time-out, the output contains everything Hyper was able to derive so far. We call this output of Hyper a partial model. In the next step of the CoRg system, the Hyper's output is used as an input to several machine learning procedures together with the unaltered natural language sentences from the tasks and the word embeddings Numberbatch from ConceptNet [24]. It is planned to generate an explanation for the given answer using the Hyper's output belonging to the chosen answer after this last step.

3 Gathering and Using Background Knowledge

During reasoning, humans naturally use background knowledge describing everyday experiences. That suggests that background knowledge plays a crucial role for cognitive computing systems as well. In the CoRg project, the following types of background knowledge are used or are planned to be used:

- lexical databases: WordNet
- first-order logic knowledge bases (like Adimen-SUMO) or first-order logic versions to higher-order knowledge bases (like SUMO, ResearchCyc, YAGO).
- knowledge graphs (ConceptNet, BabelNet[1]) and ontological knowledge bases given in RDF or OWL format.

[1] https://babelnet.org/, accessed: 11-June-2019

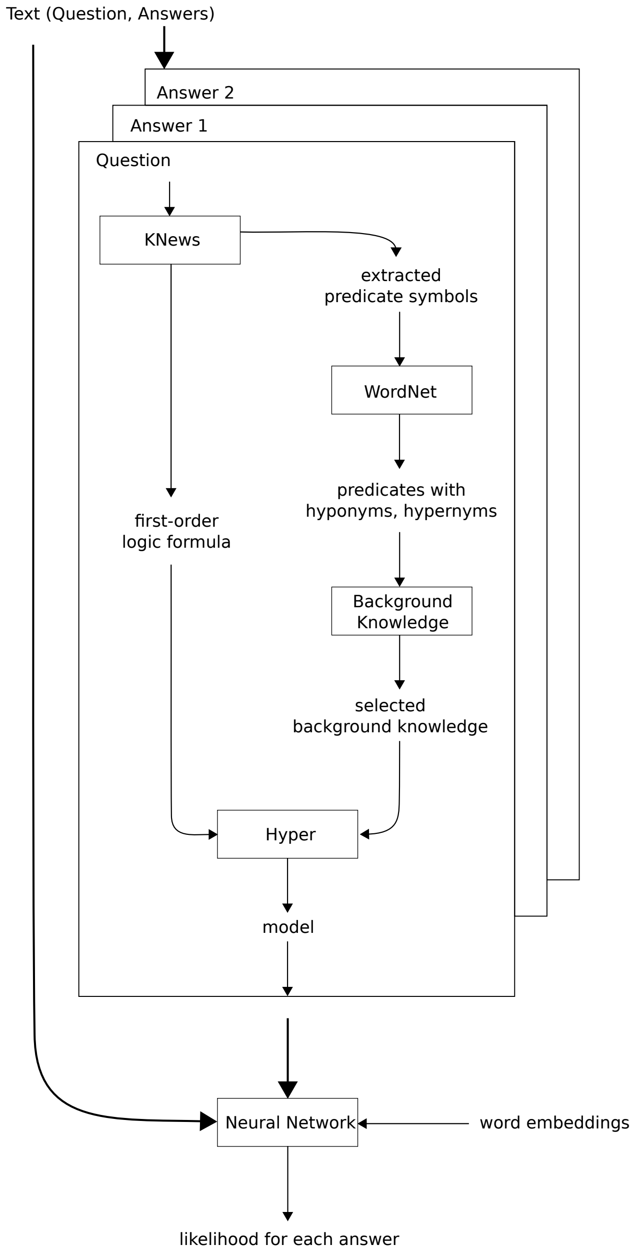

Figure 2 The CoRg system.

$$\exists x_1 \Big(thing(x_1) \wedge \exists x_2, x_3, x_4 \big(theme(x_2, x_4) \wedge actor(x_2, x_1) \wedge \\ get(x_2) \wedge outside(x_4) \wedge \\ dark(x_3) \wedge theme(x_3, x_4) \big) \Big)$$

Figure 3 First-order logic formula for the problem description of COPA problem 18 given in Fig. 1.

– word embeddings (currently from ConceptNet Numberbatch)

As the enumeration illustrates, we define the concept of background knowledge very broadly, so that knowledge bases given by different logics, knowledge graphs, lexical databases and also word embeddings are included. All sources for background knowledge are already existing sources that were not generated specifically for the task to be solved. This point is important, because although manual coding of background knowledge can produce very good results on known problems, it fails in unknown domains.

WordNet includes linguistic knowledge, and is used to specify formulae that introduce hypernyms and hyponyms of words occurring in the problem. For problem 18 in Fig. 1, KNEWS analyzes the natural language sentences and determines that “dark” belongs to the same WordNet synset as the word “night”. WordNet provides the information that “time period” is a hypernym and “weeknight” is a hyponym of “dark”. From this we generate the following formulae:

$$\forall x\, (dark(x) \leftrightarrow night(x))$$
$$\forall x\, (night(x) \rightarrow timeperiod(x))$$
$$\forall x\, (weeknight(x) \rightarrow night(x))$$

Although this is undoubtedly important background knowledge, it does not help to solve the problem as it requires to realize that there is a stronger causal connection between dark and the visibility of the moon than between dark and the occurrence of snowflakes. However, this cannot be determined only using WordNet, since it contains only taxonomic knowledge. Therefore, additional background knowledge from other sources like first-order logic knowledge bases comes to use.

One problem that occurs when selecting and integrating background knowledge is the use of different vocabularies. For example, in SUMO the term “snowflake” does not occur. Instead the terms “ice” and “snowing” are used. Therefore formulae have to be generated which connect the terms from the COPA problem with the terms in SUMO. We call these formulae *bridging formulae*. For SUMO and many other knowledge bases, a mapping of the terms used in the knowledge base to WordNet synsets exists. This mapping can be used to create the bridging formulae. For example, the WordNet mapping in SUMO contains the information that the WordNet synset “snowflake” is a subclass of the SUMO terms “ice” and “snowing”. From this, we generate the following bridging formula:

$$\forall x\, (snowflake(x) \rightarrow instance(x, ice)$$
$$\forall x\, (snowflake(x) \rightarrow instance(x, snowing)$$

Another problem with the use of existing background knowledge bases is the size of these knowledge bases: SUMO, ResearchCyc and YAGO are too large to be fully processed by an automated theorem prover. Therefore, for each COPA problem, only that part of the knowledge base that contains information relevant to the problem is considered. The selection of the part relevant for a problem is carried out with the help of so-called selection techniques. For instance, SInE (SUMO Interface Engine) [10] is a partitioning method used by many theorem provers. Given a large theory and a conjecture, SInE first determines frequencies of symbols in the knowledge base and then uses this information to select axioms which are relevant to answer the conjecture. By turning the bridging formulae and the formulae generated from WordNet into a conjecture, SInE can be used to select background knowledge relevant for the COPA problem at hand. More precisely, the CoRg system will use a SInE variant enriched with word embeddings. This leads to the fact that in our example not only knowledge is selected for the term "night" but also for terms which are similar to "night" like "after sunset". To determine this similarity, the ConceptNet Numberbatch word embeddings are used.

In addition to first-order logic knowledge bases and WordNet, knowledge graphs will be used as background knowledge. The first knowledge graph to be used explicitly (i.e. not only within Numberbatch) will be ConcepNet.[2] Specifically, it is planned to use ConceptNet to enrich the selected background knowledge. Referring to COPA problem 18 shown in Fig. 1, the knowledge selected with SInE contains information about the terms *moon* and *nighttime*, but there is no connection between these terms in the selected formulae. ConceptNet, on the other hand, contains the following triple:

moon – related to – nighttime

We plan to generate formulae from ConceptNet triples connecting terms in the selected background knowledge. In the above example, we will generate the formula

$$\forall x\ (moon(x) \rightarrow \exists y\ (relatedTo(x,y) \wedge nighttime(y)))$$

which connects the terms *moon* and *nighttime*.

All in all, the background knowledge for a given problem consists of the formulae generated from WordNet, the bridging formulae, the formulae selected from the first-order logic knowledge bases, and the formulae generated from ConceptNet. Note that this collection of background knowledge is done for each problem only once. This knowledge is combined and passed to the automated theorem prover Hyper together with the formula describing the COPA problem. Hyper is then used to perform inferences that can be found in Hyper's output. Note that we do not expect Hyper to find a proof, because the background knowledge most likely is incomplete. Hyper's output is then analyzed using machine learning techniques to determine which of the two alternative answers the inferences point to (cf. Sect. 4).

[2] http://conceptnet5.media.mit.edu, accessed: 11-June-2019

4 Reasoning with Multiple Formats

Logical reasoning alone is not sufficient to handle commonsense reasoning tasks. That is why we enhance the CoRg system with machine learning algorithms to close the gap between the derived facts and answering the commonsense tasks. Within commonsense reasoning tasks, recurrent networks with LSTM (long short-term memory) [9] are a promising strategy. They have a success rate of up to 84% in the SemEval benchmark [21]. However, since we want to use the derived facts from the logical model provided by Hyper, this approach cannot be used directly as is. Recurrent networks are used to process sequential information. While language is an example of sequential information, logical models are not. In recurrent networks the order is relevant. Thus one of the challenges to integrate neural networks into the CoRg system is the coding of the models with word embeddings such that they fit into a neural network. In the following, we briefly describe our approach with neural networks, a more detailed explanation can be found in [23].

We designed several networks with the Keras framework[3] to calculate which of the answers is more likely. Each of them have multiple inputs, for each textual input (question and both answers) one network to encode them, and another one to bring them together. The output is a vector with a likelihood for each answer. Also, every encoding network has a word embedding layer using the word embedding from ConceptNet Numberbatch. The difference between the networks is in the way they encode the question and answers in the input.

In a naïve approach, we just interpret the model as a text sequence and feed it into the network. However, as the logical model is rather large (up to 1.4 million lines), the calculation time explodes. Therefore we drop all symbols which are not covered in the word embedding as well as every duplicate statement in the Hyper model. Another approach is just to count the frequencies of the symbols in the models. This will greatly reduce the input size. Also, we can feed the original unaltered text of the benchmark set (question and two answer candidates) into an additional triplet of input layers. This will be done optionally to evaluate whether the logical model can improve the accuracy. As this input is sequential information we can use the state-of-the-art recurrent networks for these additional layers.

Putting the different parts of the system from Sect. 2 and Fig. 2 together, the computation times for one sentence (e.g. the question part) on a regular laptop are as follows:

[3] https://keras.io/, accessed: 11-June-2019

system part	time in seconds
KNews	6–9
WordNet	0.5–5.5
Bridging formula	0.02
Selection of background knowledge	1.46
Hyper	< 0.1
Likelihood computation	< 0.1

Note that Hyper is started with a timeout of 60 seconds, however Hyper rarely runs into a timeout. Also, there are multiple not recurring tasks which occur only once per run: The loading of word embeddings, e.g., takes around 30 seconds. All in all, the preparation of the models for one task (all three sentences) without one-time loading tasks takes approximately 30–40 seconds. Training the network once with the 500 logical models of the training set takes 7–8.5 minutes, while the prediction of 500 examples of the test set takes around 2 seconds.

5 Finding Appropriate Logics for Cognitive Reasoning

Cognitive science research lists various problem sets together with data from experiments on human performance in reasoning tasks. To model the various aspects of human reasoning observed in these experiments, other logics besides first-order logic can be used. The difference in the performance of humans and computers for some reasoning tasks can be explained with mental models [4,11]. For causal reasoning, mental models can be represented as sets or lists [12].

One important aspect where human reasoning differs from classical logic is the fact that we often derive conclusions under the assumption that nothing abnormal is known, i.e., that we do not have evidence that the conclusion is false. In consequence, human reasoning often is defeasible. It happens frequently that contrary evidence defeats our earlier reasoning. We intend to model these non-monotonic and defeasible aspects of knowledge by using commonsense non-monotonic reasoning techniques. In this context, first defeasible logic can be considered. It allows us to compare arguments with respect to the specificity criterion. From the preceding RatioLog project [8], an implementation of specificity is available [27]. Second, when modeling human reasoning, we have to act on the assumption of both incomplete and inconsistent knowledge. Therefore, ranking theories of knowledge from certain to vague knowledge [25] can also be taken into consideration here.

Within the TriangleCOPA challenge, norms play an important role, because humans normally expect their fellow human beings to comply with certain social norms. Therefore, normative knowledge can be used to model knowledge about met and unmet expectations as well. The study of logical systems for formalizing normative statements is called deontic logic. There are a lot of approaches in the deontic logic literature concerning formalisms for modeling normative knowledge [5]. We plan to make use of such a formalism to formalize normative statements and knowledge about expectations.

The combination of normative, defeasible, and classical reasoning shall be investigated further. For a starting point, see [20]. The Hyper theorem prover can be used to reason on commonsense reasoning problems together with first-order logic background knowledge. Since this background knowledge usually is incomplete, it is likely that for most examples the prover will not be able to find a proof. Nevertheless, output of the prover can contain valuable information when trying to construct an answer. Therefore, we will consider additional techniques to determine an answer from the output of the prover. One possibility to deal with the fact that the prover will not provide a proof, is to work with the models the prover constructed,which we already do. For satisfiable problems, Hyper is able to generate models, as mentioned earlier. To determine the right answer from the models, machine learning techniques are applicable.

6 Drawing Explainable Conclusions

The system developed in the CoRg project will not only provide a yes or no answer. In order to make the answers comprehensible, it is planned to present a justification for the given answer. As shown in Fig. 2, the more plausible alternative is determined based on the (possibly partial) models generated by the theorem prover using machine learning. The model associated with the chosen answer represents statements that could be inferred by the theorem prover. Together with the formulae used by the theorem prover during model creation, this represents valuable information. We plan to use word embeddings to first determine the statements in the model that are close to the selected answer and then extract the formulae used to derive these statements. From these formulae we then try to generate an explanation for the decision made.

7 Lessons Learnt so far

Insufficient Background Knowledge. Initial experiments using WordNet and the TPTP version of OpenCyc have shown that in many cases not enough relevant background knowledge is available. Since theorem provers can only make their inferences on the basis of background knowledge, the result is that the inferences of the theorem provers are not very helpful for solving the COPA problem. We try to solve this issue by using multiple sources of background knowledge. To do this, we are currently working on the inclusion of SUMO and Adimen-SUMO as background knowledge and

```
Karen was assigned a roommate her first year of
college. Her roommate asked her to go to a nearby
city for a concert. Karen agreed happily. The show
was absolutely exhilarating.
A1: Karen became good friends with her roommate.
A2: Karen hated her roommate.
```

Figure 4 A Story Cloze Test example.

will investigate whether it is beneficial to include ResearchCyc, YAGO, BabelNet, and CausalNet[4] [15].

Too few training data. Working with neural networks requires a huge amount of training samples, however in the COPA benchmark set can only make use of 500 examples as training set. There should be more training examples. To cope with this problem we want to integrate the Story Cloze Test with the ROCStories Corpora [18]. One ROCStory is a five-sentence story, while the associated Story Cloze Test takes the first four sentences as the given situation, where the logical consequence has to be predicted from two options. The right choice is the original fifth sentence from the ROCStory. An example of a Story Cloze Test is given in Fig. 4. Currently there exist 98,159 ROCStories and 3,744 Story Cloze Tests. The advantage of this dataset is the similarity in structure and dependency between question and answer to the COPA benchmark set, making it easy to adapt and use the CoRg system.

Issues with Matching Symbols. Sometimes a symbol from the problem does not appear in the background knowledge. In these cases, no suitable background knowledge can be selected, which in turn means that Hyper cannot perform any inferences. This problem can be avoided by selecting knowledge about symbols similar to the symbols used in the conjecture when selecting background knowledge. In order to achieve this, we have developed a version of SInE that includes word embedding. Experiments with this extended SInE version are still pending.

8 Conclusion and Future Work

In this paper, we presented our work in progress in the project CoRg – Cognitive Reasoning. Cognitive reasoning aims at modeling the human thinking and reasoning process. We evaluate our system using a commonsense reasoning benchmark set. We integrate various large knowledge bases into the system which serve as the information source for logical reasoning. Given the large size, because of the incompleteness and inconsistency of those knowledge bases, it is difficult to derive a complete proof with relying on the logical derivation processes alone.

To close the gap between the derived facts and answering the commonsense reasoning task, we additionally apply machine learning algorithms. Machine learning performs well on those benchmarks but lack the ability to properly explain the reasoning process. By combining the logical and machine learning techniques we hope to build both a well performing and explainable artificial intelligence.

So far we implemented a first prototype of the CoRg system, using KNEWS, WordNet, Hyper, ConceptNet Numberbatch and neural network techniques and evaluated it on COPA. Currently we are working on integrating Adimen-SUMO and SUMO as background knowledge. The final system will also make use of other ontologies such as ResearchCyc, YAGO and ConceptNet. These different sources of background knowledge will be designed as modules, for easy interchangeability and thus performance comparability. Furthermore we are working on enhancing our machine learning techniques to efficiently encode our logical models for the use in neural networks. To adress explanability in the neural network part, we plan to integrate a variant of the symbolic networks introduced in [6].

References

1. Álvez, J., Lucio, P., Rigau, G.: Adimen-SUMO: Reengineering an ontology for first-order reasoning. Int. J. Semantic Web Inf. Syst. **8**(4), 80–116 (2012). URL https://doi.org/10.4018/jswis.2012100105
2. Basile, V., Cabrio, E., Schon, C.: KNEWS: Using logical and lexical semantics to extract knowledge from natural language. In: Proceedings of the European Conference on Artificial Intelligence (ECAI) 2016 conference (2016)
3. Bender, M., Pelzer, B., Schon, C.: System description: E-KRHyper 1.4. In: International Conference on Automated Deduction, pp. 126–134. Springer (2013)
4. Byrne, R.M.J., Johnson-Laird, P.N.: 'if' and the problems of conditional reasoning. Trends in Cognitive Sciences **13**, 282–287 (2009)
5. Cariani, F., Grossi, D., Meheus, J., Parent, X. (eds.): Deontic Logic and Normative Systems – 12th International Conference, DEON 2014, Ghent, Belgium, Proceedings, LNAI 8554. Springer (2014). URL http://dx.doi.org/10.1007/978-3-319-08615-6
6. D'Avila Garcez, A.S., Broda, K., Gabbay, D.M.: Symbolic knowledge extraction from trained neural networks: A sound approach. Artificial Intelligence **125**(1-2), 155–207 (2001)
7. Furbach, U., Schon, C.: Commonsense reasoning meets theorem proving. In: M. Klusch, R. Unland, O. Shehory, A. Pokahr, S. Ahrndt (eds.) Multiagent System Technologies - 14th German Conference, MATES 2016, Klagenfurt, Österreich, September 27-30, 2016. Proceedings, *Lecture Notes in Computer Science*, vol. 9872, pp. 3–17. Springer (2016). URL https://doi.org/10.1007/978-3-319-45889-2_1
8. Furbach, U., Schon, C., Stolzenburg, F., Weis, K.H., Wirth, C.P.: The RatioLog project: Rational extensions of logical reasoning. KI **29**(3), 271–277 (2015). URL http://link.springer.com/article/10.1007/s13218-015-0377-9
9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997). URL http://doi.org/10.1162/neco.1997.9.8.1735

[4] https://cs-zyluo.github.io/CausalNet/, accessed: 11-June-2019

10. Hoder, K., Voronkov, A.: Sine qua non for large theory reasoning. In: N. Bjørner, V. Sofronie-Stokkermans (eds.) Automated Deduction – CADE-23, *Lecture Notes in Computer Science*, vol. 6803, pp. 299–314. Springer Berlin Heidelberg (2011). URL http://dx.doi.org/10.1007/978-3-642-22438-6_23
11. Johnson-Laird, P.N.: Mental Models: Towards a Cognitive Science of Language, Inference, and Consciousness. Cambridge University Press, Cambridge, UK (1983)
12. Khemlani, S.S., Barbey, A.K., Johnson-Laird, P.N.: Causal reasoning with mental models. Frontiers in human neuroscience **8**, 849 (2014)
13. Lenat, D.B.: Cyc: A large-scale investment in knowledge infrastructure. Communications of the ACM **38**(11), 33–38 (1995)
14. Levesque, H.J.: 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). URL http://www.aaai.org/ocs/index.php/SSS/SSS11/paper/view/2502
15. Luo, Z., Sha, Y., Zhu, K.Q., won Hwang, S., Wang, Z.: Commonsense causal reasoning between short texts. In: Proc. of 15th Int. Conf. on Principles of Knowledge Representation and Reasonging (KR'2016). Cape Town, South Africa (2016)
16. Maslan, N., Roemmele, M., Gordon, A.S.: One hundred challenge problems for logical formalizations of commonsense psychology. In: Twelfth International Symposium on Logical Formalizations of Commonsense Reasoning, Stanford, CA (2015)
17. Miller, G.A.: WordNet: a lexical database for english. Communications of the ACM **38**(11), 39–41 (1995)
18. Mostafazadeh, N., Roth, M., Louis, A., Chambers, N., Allen, J.: LSDSem 2017 shared task: The story cloze test. In: Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics, pp. 46–51 (2017)
19. Niles, I., Pease, A.: Towards a standard upper ontology. In: Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001, pp. 2–9. ACM (2001)
20. Nute, D. (ed.): Defeasible Deontic Logic, *Synthese Library: Studies in Epistemology, Logic, Methodology, and Philosophy of Science*, vol. 263. Springer (1997). URL http://dx.doi.org/10.1007/978-94-015-8851-5
21. Ostermann, S., Roth, M., Modi, A., Thater, S., Pinkal, M.: SemEval-2018 task 11: Machine comprehension using commonsense knowledge. In: Proceedings of the 12th International Workshop on Semantic Evaluation, pp. 747–757 (2018)
22. Roemmele, M., Bejan, C.A., Gordon, A.S.: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In: AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning (2011)
23. Siebert, S., Schon, C., Stolzenburg, F.: Commonsense reasoning using theorem proving and machine learning. In: A. Holzinger, P. Kieseberg, E. Weippl, A.M. Tjoa (eds.) CD-MAKE 2019 – Machine Learning and Knowledge Extraction, LNCS. Springer Nature Switzerland, Canterbury, UK (2019). To appear
24. Speer, R., Chin, J., Havasi, C.: ConceptNet 5.5: An open multilingual graph of general knowledge. In: AAAI Conference on Artificial Intelligence, pp. 4444–4451 (2017). URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14972
25. Spohn, W.: The Laws of Belief: Ranking Theory and Its Philosophical Applications. Oxford University Press, Wiesbaden (2012)
26. Suchanek, F.M., Kasneci, G., Weikum, G.: YAGO: A large ontology from Wikipedia and WordNet. Web Semant. **6**(3), 203–217 (2008). URL http://dx.doi.org/10.1016/j.websem.2008.06.001
27. Wirth, C.P., Stolzenburg, F.: A series of revisions of David Poole's specificity. Annals of Mathematics and Artificial Intelligence **78**(3), 205–258 (2016). URL http://dx.doi.org/10.1007/s10472-015-9471-9. Special issue on Belief Change and Argumentation in Multi-Agent Scenarios. Issue editors: Jürgen Dix, Sven Ove Hansson, Gabriele Kern-Isberner, Guillermo Simari