Kalpa Publications in Computing

Volume 10, 2019, Pages 20–26

Selected Student Contributions and
Workshop Papers of LuxLogAI 2018

CoRg: Commonsense Reasoning Using a Theorem Prover and Machine Learning

Sophie Siebert and Frieder Stolzenburg

Harz University of Applied Sciences, Department of Automation and Computer Sciences
Friedrichstraße 57-59, 38855 Wernigerode, Germany
{ssiebert,fstolzenburg}@hs-harz.de

Abstract

Commonsense reasoning is an everyday task that is intuitive for humans but hard to implement for computers. It requires large knowledge bases to get the required data from, although this data is still incomplete or even inconsistent. While machine learning algorithms perform rather well on these tasks, the reasoning process remains a black box. To close this gap, our system CoRg aims to build an explainable and well-performing system, which consists of both an explainable deductive derivation process and a machine learning part. We conduct our experiments on the Copa question-answering benchmark using the ontologies WordNet, Adimen-SUMO, and ConceptNet. The knowledge is fed into the theorem prover Hyper and in the end the conducted models will be analyzed using machine learning algorithms, to derive the most probable answer.

1 Introduction

Many problems humans face in their everyday life cannot be solved by solely relying on explicitly given facts, but need commonsense knowledge. The respective facts are often not formalized in knowledge bases, but can be derived indirectly, e.g., from text passages. However, this results in incomplete and inconsistent knowledge. In recent research, many scientists address commonsense reasoning tasks using primarily information retrieval, machine learning or neural network techniques [17]. While these techniques work rather well on benchmarks, the overall procedure remains a black box, unable to explain how the correct answer is derived. In our work, we address this shortcoming and develop an explainable and responsible artificial intelligence system by integrating a theorem prover into the reasoning process. The theorem prover provides a fundamental basis for the derivation process, as well as making it possible to properly explain it, and ultimately making an attempt to explain human reasoning.

Our research goal within the CoRg project (Cognitive Reasoning), conducted together with the university in Koblenz, is to evaluate whether a theorem prover can improve reasoning tasks in a prototypical system. Furthermore we will evaluate different ontologies, theorem provers and machine learning algorithms. To address incompleteness and inconsistency of knowledge, we will also consider the combination of defeasible and normative reasoning [22].

C. Benzmüller, X. Parent and A. Steen (eds.), LuxLogAI 2018 (Kalpa Publications in Computing, vol. 10), pp. 20–26

2 Commonsense Reasoning and Related Work

Commonsense reasoning is the sort of everyday reasoning humans typically perform about the world [15]. It allows us to derive knowledge about continuity and object permanence, e.g., if a person enters a room, then afterwards, the person will be in the room, if she has not left the room. We have knowledge about objects, events, space, time, and mental states and may use that knowledge. All this implicit background knowledge is part of everyday human reasoning and must be added to a cognitively adequate automated reasoning system.

In current commonsense reasoning tasks such as the SemEval competition 2018 Task 11 [17], the participating systems already scored with an accuracy of up to 84%. These tasks consist of a long text (i.e., 10-20 short sentences) describing a situation and related questions with two possible answers, respectively. To solve these tasks, nearly all participants used recurrent neural networks with LSTM (long short-term memory) [8], which perform well on sequential knowledge (e.g. a sequence of words). Furthermore, most of them used word embeddings such as GloVe [18], which map words to vectors of real numbers, embedding from a space with one dimension per word to a vector space with a much lower dimension, such that relative similarities correlate with semantic similarity. Some systems also used the knowledge graph ConceptNet [12]. However, these approaches remain a black box and the derivation process cannot be explained.

Nowadays there is a chance to join automated deduction and commonsense reasoning [6], namely within the paradigm of cognitive computing, which allows the implementation of rational reasoning [10]. This has been done, e.g., in LogAnswer and RatioLog [5, 7]. The corresponding web system is an open-domain question answering system, accessible via a web interface similar to that of a search engine. The knowledge used to answer the question is gained from 29.1 million natural-language sentences of a snapshot of the German Wikipedia.

3 The Structure of CoRg

The goal of the CoRg system is to successfully complete a question-answering task in the field of commonsense reasoning. The tasks to solve consist of a question or situation and two possible answers in natural language as shown in Fig. 2. This information is fed into the CoRg system, which consists of several components depicted in Fig. 1. First, the text of the question and answers will be transformed into a logical representation using KNEWS [2]. During this process the keywords from the text will be extracted and looked up from WordNet [13], providing us with the WordNet IDs. Those are used to lookup the WordNet synset, i.e., a grouped set of cognitive synonyms, to expand the knowledge base with hyponyms and hypernyms. Based on the original predicates from KNEWS and the added hyponyms and hypernyms, we search for other relevant words from different ontologies. After gathering all relevant information, we execute the theorem prover Hyper [4] on it. It determines a possible model from the given logical knowledge. We will use this model as an input to sev-

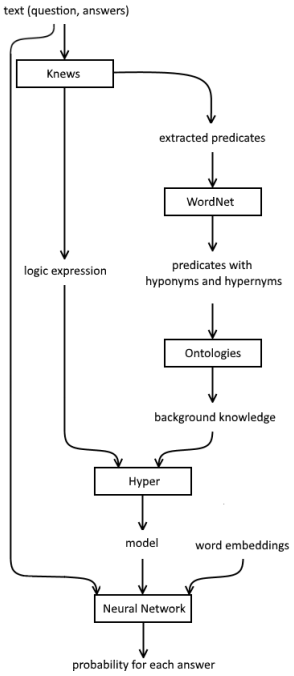

Figure 1: The CoRg System

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

Figure 2: Problem 1 from the Copa benchmark set.

eral machine learning algorithms, to calculate the most likely answer.

3.1 Copa

We currently use the Copa benchmark set (Choice of Plausible Alternatives) [19] to evaluate the system. Nevertheless, every benchmark with a similar structure can be used within the CoRg system. The Copa benchmark set consists of 1,000 tasks. Each task consists of a question or situation and two possible answer candidates, as shown in Fig. 2. The set is evenly split into cause and result categories, which refers to the relationship between the questions and answer. The cause category needs backward causal reasoning while the result category requires forward causal reasoning.

3.2 Ontologies

To successfully complete a question-answering task, background knowledge is required. In the CoRg system, we currently use WordNet and Adimen-SUMO as well as the word embeddings from Numberbatch [20]. Later on, we also want to integrate SUMO, ResearchCYC [11], YAGO [21], and ConceptNet. WordNet [13] is a lexical database, whose IDs are used as links between the other knowledge bases. Also it provides the hyponyms and hypernyms of the original words from the task. Taxonomic knowledge however is not sufficient to solve reasoning tasks, that is why we also need sources like first-order knowledge bases. Adimen-SUMO [1] is such a database. It is a revised version of SUMO [16], which is one of the largest formal public ontologies with WordNet links.

When integrating different knowledge bases, a variety of problems can occur. First, the vocabulary can be different. However, most of the knowledge bases have a WordNet synset mapping, that allows us to unify all knowledge bases using WordNet. Second, the knowledge bases are large, with lots of uninteresting information concerning the task. To select only relevant information, SInE (SUMO Interface Engine) [9] is used, which determines relevance by considering symbol frequencies. By means of this, we built three knowledge domains for each of the question and answers. The distance or links between these domains can help us find the right answer to the task. Later, we will add ConceptNet [12] to find edges between the domains to connect them. ConceptNet is a knowledge graph with associative and everyday knowledge and gives us the possibility to think outside the box.

3.3 The Theorem Prover Hyper

After we gathered the relevant knowledge from various sources we execute the theorem prover Hyper [4] on the tasks. Hyper is an automated theorem prover based on the E-Hyper tableau calculus [3]. For each task Hyper derives a model for the question and the two answer candidates, providing us with three models per task, one for each text.

4 Machine Learning

As stated in Sect. 2, machine learning, especially neural networks are state-of-the-art, when solving commonsense reasoning tasks. Other than the participants of the SemEval task using recurrent neural networks, we use additional information in the form of a first-order logic model. Recurrent networks perform well on sequential data, as they remember early input and the order. In a logical model however the input order is irrelevant, because it consist of a conjunction of true statements. In addition, the models often contain millions of entries, which are often redundant. The challenge is to encode the logical models in a suitable manner, such that a neural network can efficiently process them. In the following different approaches are described to process the tasks and models to calculate a probable answer.

Naive Euclidean Distance. A first baseline approach to screen the data is a simple calculation of the Euclidean distance of the question to each of the answers. We used the Numberbatch word embeddings [20] to calculate a mean for each question and answers and took the difference. This approach has a accuracy of lower than 50%, showing that commonsense reasoning indeed is difficult to approach with simple information retrieval methods and similarity measures.

Feed-forward neural network with word embeddings. The next approach was to use a multi-input feed-forward neural network. Each task is taken as a training example. The neural network itself consist of three input networks, one network for the question and one for each of the answers. The models are filtered for the Numberbatch word embeddings, e.g., every word, which does not appear in the word embedding is thrown away. The remaining words were put in a sequence and fed into the network. The results were at exactly 50% and the computing time exploded, as the input was very long.

Encode models with frequencies. To keep the input lower, we thought of implementing a frequency input layer, as the order of the predicates in the model do not matter anyway. We implemented a dictionary integrated into our networks and are currently working on altering the framework, so this dictionary can be processed together with the information of the word embeddings.

Encode structure of models as graphs. In addition to the predicates which are deducted, we also have information about the proof structure. This could also be integrated in the neural network by building an adjacency matrix.

Use the unaltered natural language. Every neural network option mentioned in the previous paragraphs can be advanced by feeding the neural network the unaltered natural language text from the Copa tasks. This can either be done solely or additionally, requiring another three input networks. Since in this case we work with dependent sequential information, recurrent networks should be applied like in the SemEval tasks.

5 Problems and Lessons Learned

Using the Copa data for training with its 500 training examples is by far not sufficient to train a neural network. To cope with this problem, we have to find similar problems with a much larger database. These problems should have a structure similar to Copa, so we do not have to alter

```
 Q: 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 3: A Story Cloze Test Example

the network structure. For this, the Story Cloze Test with the ROCStory Corpus [14] seems to be appropriate. In this corpus are currently 98,159 ROCStories with 3,744 Story Cloze Tests. There, one task is a short five-sentence story, where the final sentence has to be guessed from the first four as shown in Fig. 3. While they have a longer question/situation part, the Story Cloze Test can be interpreted in the same way as the Copa benchmark, with a question and two possible answers. They also address the same kind of knowledge and logical interference as the Copa tasks. Currently, we are integrating this data into the system.

Another problem is the matching of predicates in the model with the words in the word embeddings. Often the words are derived from the ontologies and have special characters or compound words in it. While we integrate new ontologies, we work on matching rules, so every word will have an external mapping to a WordNet synset, which on the other hand is what Numberbatch uses. For the remaining unmapped words we can apply multiple strategies, such as unking or substring matching. Unking means to map all unknown words to one default identifier, while substring matching tries to find similar words based on common substrings.

6 Conclusion and Future Work

Within the CoRg project, we want to analyze whether a theorem prover can improve the reasoning process of question-answering tasks in the domain of commonsense reasoning. For this, we combine different ontologies, theorem provers, and machine learning algorithms. Building interfaces and linking code between those applications is a challenge here, e.g., to match the predicates from the texts and ontologies with the word embedding for machine learning, as well as finding an appropriate way to encode those large models from the theorem prover to fit into machine learning. Also tiny datasets need some pre-training using slightly different larger train sets.

In future work, we want to be able to integrate different modules into our system, such that ontologies, theorem provers and machine learning algorithms are interchangeable to see their impact on the outcome. Also we want to add deontic and defeasible logic. The work in progress stated in this paper will be continued to improve the accuracy of the procedure to significantly more than 50%. Ultimately, we might be able to reverse engineer the derivation process, which will give us significant insights on human reasoning.

Acknowledgements

We would like to thank Ulrich Furbach, Claudia Schon, and Christian Schmittel for helpful discussions, comments, hints, or implementational work. The research reported in this paper has been supported by the German Research Foundation (DFG) under grant STO 421/8–1 within the project CoRg – Cognitive Reasoning.

References

[1] Javier Álvez, Paqui Lucio, and German Rigau. Adimen-SUMO: Reengineering an ontology for first-order reasoning. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(4):80–116, 2012.

[2] Valerio Basile, Elena Cabrio, and Claudia Schon. 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] Peter Baumgartner, Ulrich Furbach, and Ilkka Niemelä. Hyper tableaux. In *European Workshop on Logics in Artificial Intelligence*, pages 1–17. Springer, 1996.

[4] Markus Bender, Björn Pelzer, and Claudia Schon. System description: E-KRHyper 1.4. In *International Conference on Automated Deduction*, pages 126–134. Springer, 2013.

[5] Ulrich Furbach, Ingo Glöckner, and Björn Pelzer. An application of automated reasoning in natural-language question answering. *AI Communications*, 23:241–265, 2010.

[6] Ulrich Furbach and Claudia Schon. Commonsense reasoning meets theorem proving. In Matthias Klusch, Rainer Unland, Onn Shehory, Alexander Pokahr, and Sebastian Ahrndt, editors, *Multiagent System Technologies*, pages 3–17, Cham, 2016. Springer International Publishing.

[7] Ulrich Furbach, Claudia Schon, Frieder Stolzenburg, Karl-Heinz Weis, and Claus-Peter Wirth. The RatioLog project: Rational extensions of logical reasoning. *KI*, 29(3):271–277, 2015.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[9] Kryštof Hoder and Andrei Voronkov. Sine qua non for large theory reasoning. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *Automated Deduction – CADE-23*, volume 6803 of *Lecture Notes in Computer Science*, pages 299–314. Springer Berlin Heidelberg, 2011.

[10] John E. Kelly III and Steve Hamm. *Smart Machines: IBM's Watson and the Era of Cognitive Computing*. Columbia Business School Publishing, 2013.

[11] Douglas B. Lenat. CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995.

[12] Hugo Liu and Push Singh. ConceptNet – a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226, 2004.

[13] George A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[14] Nasrin Mostafazadeh, Michael Roth, Annie Louis, Nathanael Chambers, and James Allen. LSDSem 2017 shared task: The story cloze test. In *Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics*, pages 46–51, 2017.

[15] Erik T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann, San Francisco, 2nd edition, 2014.

[16] Ian Niles and Adam 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.

[17] Simon Ostermann, Michael Roth, Ashutosh Modi, Stefan Thater, and Manfred Pinkal. SemEval-2018 task 11: Machine comprehension using commonsense knowledge. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 747–757, 2018.

[18] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[19] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, pages 90–95, 2011.

[20] Robyn Speer, Joshua Chin, and Catherine Havasi. ConceptNet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*, pages 4444–4451, 2017.

[21] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A large ontology from Wikipedia and WordNet. *Web Semant.*, 6(3):203–217, September 2008.

[22] Claus-Peter Wirth and Frieder Stolzenburg. A series of revisions of David Poole's specificity. *Annals of Mathematics and Artificial Intelligence*, 78(3):205–258, 2016. Special issue on Belief Change and Argumentation in Multi-Agent Scenarios. Issue editors: Jürgen Dix, Sven Ove Hansson, Gabriele Kern-Isberner, Guillermo Simari.