# Ontology Reasoning with Deep Neural Networks (Extended Abstract)*

**Patrick Hohenecker**[1,2] and **Thomas Lukasiewicz**[1,2]
[1]Department of Computer Science, University of Oxford, UK
[2]Serein AI, London, UK
{patrick.hohenecker, thomas.lukasiewicz}@cs.ox.ac.uk

## Abstract

The ability to conduct logical reasoning is a fundamental aspect of intelligent human behavior, and thus an important problem along the way to human-level artificial intelligence. Traditionally, logic-based symbolic methods from the field of knowledge representation and reasoning have been used to equip agents with capabilities that resemble human logical reasoning qualities. More recently, however, there has been an increasing interest in using machine learning rather than logic-based symbolic formalisms to tackle these tasks. In this paper, we employ state-of-the-art methods for training deep neural networks to devise a novel model that is able to learn how to effectively perform logical reasoning in the form of basic ontology reasoning.

## 1 Introduction

Implementing human-like logical reasoning has been among the major goals of artificial intelligence (AI) research ever since, and has recently also enjoyed increasing attention in the field of machine learning (ML). However, a noticeable commonality of previous approaches in this area is that they, with a few very recent exceptions [Serafini and d'Avila Garcez, 2016; Cai *et al.*, 2017; Rocktäschel and Riedel, 2017; Cingillioglu and Russo, 2018; Dai *et al.*, 2018; Evans *et al.*, 2018; Manhaeve *et al.*, 2018], entertain a quite informal notion of reasoning, which is often simply identified with a particular kind of prediction task. This contrasts the (traditional) understanding of reasoning as an application of mathematical proof theory, like it is used in the context of logic-based knowledge representation and reasoning (KRR). Interestingly, however, it can be observed that, under certain provisions, even the best reasoning models based on ML are still not in a position to compete with their symbolic counterparts. To close this gap between learning-based and KRR methods, we introduce a novel model architecture, called recursive reasoning network (RRN), which makes use of recent advances in the area of deep neural networks [Bengio, 2009]. By design, this model is much closer to logic-based symbolic methods than most of the other learning-based approaches, but the fact that it employs ML allows for overcoming many obstacles that we encounter with KRR methods in practice.

**Ontology:** `human(X)` ← `holds(X,_)`.
*Only humans can hold things.*
`object(Y)` ← `holds(_,Y)`.
*Only objects can be held.*
⊥ ← `object(X)` ∧ `human(X)`.
*Objects are not human beings and vice versa.*
`isAt(Y,Z)` ← `holds(X,Y)` ∧ `isAt(X,Z)`.
*Objects are at the same location as the one holding them.*
⊥ ← `isAt(X,Y)` ∧ `isAt(X,Z)` ∧ `Y≠Z`.
*Nobody/nothing can be at two locations at the same time.*

**Facts:** `holds(mary,apple)`, `isAt(mary,kitchen)`.

**Queries:** `?human(apple)`
*Is the apple human?* → **false**
`?isAt(apple,kitchen)`
*Is the apple in the kitchen?* → **true**
`?isAt(mary,bedroom)`
*Is Mary in the bedroom?* → **false**

Figure 1: A simple example of an ontology (inspired by the well-known bAbI tasks [Weston *et al.*, 2015]) that describes a few rules for reasoning over human beings, objects, and their locations. Combined with the stated facts, it allows for answering queries like »*Is the apple a human being?*« or »*Is Mary in the bedroom?*«.

Ontology reasoning refers to a common scenario where the inference rules to be used for reasoning, called the ontology in this context, are specified alongside the factual information that we seek to reason about. (Fig. 1 provides an example of this setting.) The rationale behind this separation of ontology and facts is that it allows for adopting the same set of rules for reasoning about different, independent data. One may ask why we would like to set about this problem by means of ML in the first place. Most KRR formalisms that are used for reasoning today are rooted in symbolic logic, and thus, as mentioned above, employ mathematical proof theory to answer queries about a given problem. However, while this, in theory, allows for answering any kind of (decidable) question accurately, most of these approaches suffer from a number of issues in practice, like difficulties with handling incomplete, conflicting, or uncertain data, to name just a few. In contrast to this, ML models are often highly scalable, more resistant to disturbances in the data, and capable of providing predictions even if the formal effort fails. By using state-of-the-art techniques of deep learning, we aim to manage the balancing act between approximating the highly desirable (theoretical)

*Extended abstract of an article in the Journal of Artificial Intelligence Research [Hohenecker and Lukasiewicz, 2020].

properties of the formal approach, on the one hand, and utilizing the robustness of ML, on the other.

The main contributions of this paper are briefly as follows.

- We present a novel deep neural architecture for a model that is able to effectively perform logical reasoning in the form of basic ontology reasoning.
- We present and make freely available several very large, diverse, and challenging datasets for learning and benchmarking ML approaches to basic ontology reasoning.
- We present extensive experimental evaluations on these benchmarks, which show that our model is able to learn to perform highly accurate ontology reasoning.

## 2 The Recursive Reasoning Network (RRN)

With the introduction of RRNs, we replace formal ontology reasoning with computing a learned deep neural network to remedy the issues outlined above. Thereby, following the spirit of the considered problem, every RRN is trained relative to a particular ontology, and thus, like the formal counterpart, independent of the specific facts that it is provided with. To that end, the vocabulary of classes and relations used by the ontology determines the recursive layers that are available in an RRN, and hence the structure of the same. In contrast to this, however, the rules that are used for reasoning are not provided to the model directly, but have to be learned from the training data. When a trained model is applied to a particular set of facts, then, on the face of it, it operates in two stages: first, it generates vector representations (so-called embeddings) for all individuals that appear in the considered data, and second, it computes predictions for queries solely based on these generated vectors.

RRNs are based on the idea that we can encode all the available information about an individual, both specified and inferable, in its embedding. A similar idea is used, for example, in the context of natural language processing, where real vectors are used to represent the meaning of text [Mikolov *et al.*, 2013]. Given a set of facts, specified as triples, we start by randomly generating initial embeddings for all the individuals that appear in any of them. After this, the model iterates over all the triples, and, for each of them, updates the embeddings of the individuals involved. Any such update is supposed to store the considered triple in the individuals' embeddings for one thing, but also to encode possible inferences based on the same. So, intuitively, a single update step conducts local reasoning based on what is encoded in the embeddings already as well as on the new information that was gained through the provided fact. An obvious necessity implied by this local reasoning scheme is that the model, in general, has to sift through all the data multiple times. This is essential in order to allow for multi-step, also called multi-hop, reasoning, which is based on several triples at the same time, since information might need to transpire from one individual's embedding to another's. The actual number of iterations required depends on the respective dataset, though.

From a technical perspective, the outlined procedure for generating embeddings corresponds to computing a recursive neural network [Pollack, 1990] that receives the randomly generated initial embeddings as input and provides the final embeddings as output. The structure of the network depends on the set of facts being processed, and its recursive layers compute the update operations described above, which is why we refer to them as update layers.

Once the desired embeddings are generated, they can be used to answer atomic queries about the data that they are computed from. To that end, the model provides various multi-layer perceptrons (MLPs) for computing predictions about relations between two individuals and class memberships of a single individual. Notice that the only inputs provided to these MLPs are the embeddings of the individuals that are involved in a particular query, which is why the model has to ensure that all the information that is needed for answering such is effectively encoded during the first step. Thus, the second step is just needed to uncover the knowledge that is encoded in individual embeddings, while the actual reasoning happens before.

A notable characteristic of the RRN is that it performs deductive inference over entire knowledge bases, and, like its symbolic stencil, aims to encode all possible inferences in the created individual embeddings, rather than answering just a single query. Because of this, the model is able to unravel complex relationships that are hard to detect if we try to evaluate the inferability of an isolated triple of interest only. Furthermore, the fact that the RRN conducts logical inference over all classes and relations simultaneously allows for leveraging interactions between any of them, and thus further adds to improving the model's predictive performance.

## 3 Evaluation

In this section, we summarize the experimental results that we obtained with the RRN on five different datasets.

### 3.1 Datasets

To assess the suggested approach, we trained and evaluated an RRN on five different datasets, two out of which were artificially created toy datasets, two were extracted from real-world knowledge bases, and one was a generated dataset based on a real-world ontology:

• **Family trees.** In this toy dataset (Fig. 2), samples describe pedigrees of different sizes such that the only facts that are available in any of them are the genders of the people involved and the immediate ancestors (i.e., parent-of relations) among them. Besides this, the used ontology specifies rules for reasoning about 28 different family relations, ranging from »*easy*« inferences (e.g., fatherhood) to more elaborate ones (e.g., being a girl first cousin once removed).

• **Countries.** The second toy dataset is based on the countries knowledge base [Bouchard *et al.*, 2015], which describes the adjacency of countries together with their locations in terms of regions and subregions. In every sample knowledge base, some of the countries' regions and subregions, respectively, are not stated as facts, but supposed to be inferred from the information that is provided about their neighborhoods. Following Nickel *et al.* [2016], we constructed three different versions of the dataset: S1 (easy), S2, and S3 (hard). An interesting characteristic of the latter two versions of the dataset

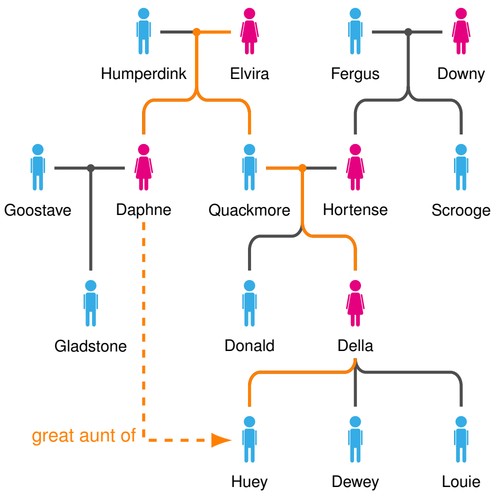

Figure 2: An example of a family tree, like it is found in the first toy dataset that the RRN model was tested on. While the genders of all people involved as well as the immediate relations between parents and their children are provided as facts, all other family relations have to be inferred. The figure depicts an example of a four-hop inference, namely, Daphne being a great aunt of Huey.

is that the sample knowledge bases are constructed such that parts of the missing relations cannot be inferred by means of the ontology at all, which challenges the model's ability to generalize beyond pure ontology reasoning.

- **DBpedia.** This dataset was extracted from the well-known DBpedia knowledge base [Bizer *et al.*, 2009], which represents part of the knowledge that is available in terms of natural language on Wikipedia. As this data does not naturally separate into samples, we extracted sample knowledge bases that are subgraphs of the original knowledge graph such that each of them contains a total of 200 individuals. DBpedia employs a massive vocabulary consisting of thousands of classes and relations, respectively. To make the according experiments more computationally feasible, we restricted this to the 101 most frequent classes and those 518 relation types that allow for the greatest numbers of inferences.

- **Claros.** This data was extracted from Claros, which is a formalization of a catalog of archaeologic artifacts [Nenov *et al.*, 2015]. Like for DBpedia, we extracted sample knowledge bases over 200 individuals each. However, we used all the Claros ontology, consisting of 33 classes and 77 relations.

- **UMLS reasoning.** In addition to this, we created a dataset based on the Unified Medical Language System (UMLS) [McCray, 2003], an ontology that has been introduced for describing concepts from the biomedical domain in a uniform way by means of 127 classes and 53 relations. Interestingly, UMLS is commonly used as a benchmark for methods of knowledge graph completion [Kok and Domingos, 2007]. In this context, however, UMLS is not interpreted as an ontology, but simply viewed as an ordinary knowledge graph. To avoid confusion, we use the term »*UMLS-reasoning*« to refer to our own dataset in order to set it apart from the benchmark by Kok and Domingos [2007].

**Classes**

| **Dataset** | **Facts** | | **Inferences** | |
|---|---|---|---|---|
| | Acc. | $F_1$ | Acc. | $F_1$ |
| family trees | 1.000 | 1.000 | 1.000 | 1.000 |
| countries (S1) | — | — | 1.000 | 1.000 |
| countries (S2) | — | — | 1.000 | 1.000 |
| countries (S3) | — | — | 1.000 | 1.000 |
| DBpedia | — | — | 0.998 | 0.997 |
| Claros | — | — | 0.999 | 0.999 |
| UMLS-reasoning | — | — | 0.990 | 0.994 |

**Relations**

| **Dataset** | **Facts** | | **Inferences** | |
|---|---|---|---|---|
| | Acc. | $F_1$ | Acc. | $F_1$ |
| family trees | 1.000 | 1.000 | 0.999 | 0.976 |
| countries (S1) | 1.000 | 1.000 | 0.999 | 0.999 |
| countries (S2) | 0.999 | 0.997 | 0.999 | 0.929 |
| countries (S3) | 0.999 | 0.996 | 0.999 | 0.916 |
| DBpedia | 0.998 | 0.998 | 0.989 | 0.962 |
| Claros | 0.999 | 0.999 | 0.996 | 0.997 |
| UMLS-reasoning | 0.997 | 0.996 | 0.997 | 0.989 |

Table 1: Summary of experimental results. Accuracy and $F_1$ score are reported separately for class memberships and relations, and within each group, for those triples that describe specified knowledge (i.e., facts) and those that represent inferable information.

Each of the datasets above is a collection of sample knowledge bases that share a common ontology—5000 for training, 500 for validation, and 500 for testing. During training and evaluation, the model was provided with all the facts in the considered samples, and had to compute predictions for both facts and inferences that were derivable from the same.

### 3.2 Results

Table 1 summarizes the results that have been achieved in our experiments, and there are a number of interesting aspects to notice across all the considered reasoning tasks. First, we see that the RRN is able to effectively encode provided facts, about both classes and relations, as these are predicted correctly from the created embeddings with an accuracy greater than 98.9%, and except for UMLS-reasoning, even with more than 99.8%. Furthermore, we observe that the model also learns to reason over classes with an equally high accuracy of 99.8% on all datasets except UMLS-reasoning, for which we find an accuracy of 99.0%. This difference is not surprising, though, as the UMLS ontology specifies by far the largest taxonomy of classes. For reasoning over relations, we find a slightly lower accuracy of 98.9% on DBpedia, while derivable relations in other datasets are predicted correctly in at least 99.6% of all cases. Again, however, this difference is not surprising, as DBpedia is by far the most complex among the ontologies that were used in our experiments, and might thus require a larger training dataset in order to achieve an even higher accuracy. Also, it was to be expected that the model would generally perform better in predicting inferable classes than relations, since most of these are inferences depending on single triples only.

## 4 Learning-based vs. Symbolic Methods

One important question is how to motivate the neural approach to ontology reasoning in the first place, and how the RRN relates to purely symbolic methods of logical reasoning. First, the presented model is a step towards answering the wide open problem of how to combine deep learning technologies with symbolic methods for logical reasoning, which is commonly regarded as a prerequisite for further substantial progress in AI. Implementing symbolic ontology reasoning with neural networks of very high accuracy in some sense bridges the gap between neural and symbolic methods, and offers new ways of providing ML models with reasoning capabilities that have previously been reserved for symbolic methods only, which opens up interesting new opportunities. From a ML perspective, the RRN can be considered as a method of knowledge graph embedding [Wang *et al.*, 2017] that produces semantically meaningful embeddings of the individuals in a knowledge graph. These, in turn, may serve as input to models that are used for learning downstream tasks, and so allows for leveraging symbolic background knowledge in learning deep neural networks (and so for knowledge transfer and for learning from smaller amounts of data) as well as explainable symbolic inference in computing predictions for improved explainability of the learned neural systems.

Second, the neural approach to ontology reasoning is useful in its own right as an alternative to symbolic methods to logical reasoning. Although it does not allow for fully accurate logical reasoning, it paves the way for highly scalable implementations of nearly accurate approximate logical reasoning via parallel computations on GPUs. Such implementations cannot be used for safety-critical applications (such as for verifying the control of nuclear power plants), but may be sufficiently accurate for other applications where full accuracy is not required (e.g., question answering over the web).

In the same vein, one major issue that many symbolic approaches, including all reasoning formalisms rooted in classical logic, suffer from is conflicting information. In practice, however, information is frequently imperfect, which is why conflicts inevitably have to be dealt with in many use cases that symbolic reasoning is applied to. While many reasoning methods simply do not work in any such case, our experiments with the RRN suggest that the model is able to effectively resolve conflicts and thus better suited for applying logical reasoning in a real-world scenario. Even though there exist formalisms for reasoning over inconsistent knowledge bases that are also able to resolve conflicts, such as inconsistency-tolerant reasoning [Lembo *et al.*, 2010] and paraconsistent logics [Middelburg, 2011], these approaches are generally quite limited in practice, since there is a price to be paid in terms of computational complexity.

Finally, another challenge that is commonly encountered in practice is missing information, that is, details that are neither specified as facts in the considered knowledge base nor inferable from them via symbolic reasoning. Strictly speaking, recovering such missing pieces is a prediction rather than a reasoning task, and hence usually not considered in the context of symbolic reasoning. This is not in line with the requirements that are typically faced in practice, though, as we frequently seek to do both, compute predictions and perform reasoning. Again, however, our experiments indicate that this is exactly what the RRN does. To that end, we observed that the model is able to provide sensible predictions for any missing details that are compatible with the considered set of facts relative to the used ontology in many cases, and that these, in turn, also affect inferences computed by the model.

## 5 Related Work

While there has been an increasing interest in the application of learning-based methods to various kinds of logical reasoning in the last few years, ontology reasoning in particular has received just modest attention. The only published paper that we are aware of is by Makni and Hendler [2018], who developed an approach to RDFS reasoning contemporaneously with the work presented in this article. To that end, Makni and Hendler use a BiGRU encoder-decoder architecture for encoding entire knowledge graphs as embedding vectors and subsequently decoding inference graphs from the same.

Another recent article by Ebrahimi *et al.* [2018] addresses reasoning over RDF knowledge base via an adapted version of end-to-end memory networks [Sukhbaatar *et al.*, 2015]. To that end, Ebrahimi *et al.* treat triples like text, and map the elements of any such (i.e., subject, predicate, and object) to embedding vectors. After this, the embedded triples are placed in the memory of an adapted memory network, which computes a prediction for a query that is provided as an embedded triple as well.

Apart from this, a lot of previous work addresses the combination of logic-based symbolic reasoning and deep learning in some way, but is not related to ontology reasoning per se. For an overview, see Hohenecker and Lukasiewicz [2020].

## 6 Summary and Outlook

In this work, we have introduced the RRN as a novel model architecture for deep-learning-based ontology reasoning, and presented results of an experimental evaluation of the same on numerous benchmark datasets. To that end, we have shown that the RRN learns to conduct highly accurate reasoning in a logic-based sense, and is able to work with complex real-world knowledge bases.

Despite the fact that an RRN is effectively guided by the ontology that it has been trained on, it cannot provide justifications for predictions yet. As important future work, we are thus currently extending the presented architecture such that it allows for inducing rules and explanations, such as inference graphs, alongside computed predictions, hence making the model a fully interpretable neural reasoner.

## Acknowledgments

This work was supported by the UK EPSRC grants EP/J008-346/1, EP/R013667/1, EP/L012138/1, and EP/M025268/1, and the Alan Turing Institute under the EPSRC grant EP/N510129/1. Furthermore, Patrick is supported by the EP-SRC under the grant OUCL/2016/PH and an Oxford-Deep-Mind Graduate Scholarship under the grant GAF1617_OG-SMF-DMCS_1036172. We also acknowledge the use of the EPSRC-funded Tier 2 facility JADE (EP/P020275/1).

## References

[Bengio, 2009] Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

[Bizer *et al.*, 2009] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia — a crystallization point for the web of data. *Journal of Web Semantics*, 7(3):154–165, 2009.

[Bouchard *et al.*, 2015] Guillaume Bouchard, Sameer Singh, and Theo Trouillon. On approximate reasoning capabilities of low-rank vector spaces. In *Proceedings of the 2015 AAAI Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches*, pages 6–9, 2015.

[Cai *et al.*, 2017] Cheng-Hao Cai, Dengfeng Ke, Yanyan Xu, and Kaile Su. Symbolic manipulation based on deep neural networks and its application to axiom discovery. In *Proceedings of the 2017 International Joint Conference on Neural Networks*, pages 2136–2143, 2017.

[Cingillioglu and Russo, 2018] Nuri Cingillioglu and Alessandra Russo. DeepLogic: End-to-end logical reasoning. *arXiv:1805.07433*, 2018.

[Dai *et al.*, 2018] Wang-Zhou Dai, Qiu-Ling Xu, Yang Yu, and Zhi-Hua Zhou. Tunneling neural perception and logic reasoning through abductive learning. *arXiv:1802.01173*, 2018.

[Ebrahimi *et al.*, 2018] Monireh Ebrahimi, Md Kamruzzaman Sarker, Federico Bianchi, Ning Xie, Derek Doran, and Pascal Hitzler. Reasoning over RDF knowledge bases using deep learning. *arXiv:1811.04132*, 2018.

[Evans *et al.*, 2018] Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. Can neural networks understand logical entailment? In *International Conference on Learning Representations*, 2018.

[Hohenecker and Lukasiewicz, 2020] Patrick Hohenecker and Thomas Lukasiewicz. Ontology reasoning with deep neural networks. *Journal of Artificial Intelligence Research*, 68, 2020.

[Kok and Domingos, 2007] Stanley Kok and Pedro Domingos. Statistical predicate invention. In *Proceedings of the 24th International Conference on Machine Learning*, pages 433–440, 2007.

[Lembo *et al.*, 2010] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo. Inconsistency-tolerant semantics for description logics. In *Proceedings of the 4th International Conference on Web Reasoning and Rule Systems*, pages 103–117, 2010.

[Makni and Hendler, 2018] Bassem Makni and James Hendler. Deep learning for noise-tolerant RDFS reasoning. *Semantic Web*, 10(5):823–862, 2018.

[Manhaeve *et al.*, 2018] Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. DeepProbLog: Neural probabilistic logic programming. *arXiv:1805.10872*, 2018.

[McCray, 2003] Alex T. McCray. An upper-level ontology for the biomedical domain. *Comparative and Functional Genomics*, 4:80–84, 2003.

[Middelburg, 2011] Cornelis A. Middelburg. A survey of paraconsistent logics. *arXiv:1103.4324*, 2011.

[Mikolov *et al.*, 2013] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013.

[Nenov *et al.*, 2015] Yavor Nenov, Robert Piro, Boris Motik, Ian Horrocks, Zhe Wu, and Jay Banerjee. RDFox: A highly-scalable RDF store. In *Proceedings of the 14th International Semantic Web Conference*, pages 3–20, 2015.

[Nickel *et al.*, 2016] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 1955–1961, 2016.

[Pollack, 1990] Jordan B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1/2):77–105, 1990.

[Rocktäschel and Riedel, 2017] Tim Rocktäschel and Sebastian Riedel. End-to-end differentiable proving. In *Advances in Neural Information Processing Systems 30*, pages 3788–3800, 2017.

[Serafini and d'Avila Garcez, 2016] Luciano Serafini and Artur d'Avila Garcez. Logic Tensor Networks: Deep learning and logical reasoning from data and knowledge. *arXiv:1606.04422*, 2016.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems 28*, pages 2440–2448, 2015.

[Wang *et al.*, 2017] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017.

[Weston *et al.*, 2015] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards AI-complete question answering: A set of prerequisite toy tasks. *arXiv:1502.05698*, 2015.