CLUTRR: A Diagnostic Benchmark for Inductive Reasoning from Text Appendix & Supplementary Materials

Koustuv Sinha [1,3,4], **Shagun Sodhani** [2,3], **Jin Dong** [1,3], **Joelle Pineau** [1,3,4] and **William L. Hamilton** [1,3,4]
[1] School of Computer Science, McGill University, Canada
[2] Université de Montréal, Canada
[3] Montreal Institute of Learning Algorithms (Mila), Canada
[4] Facebook AI Research (FAIR), Montreal, Canada
{koustuv.sinha, sshagunsodhani, jin.dong, jpineau, wlh}
@{mail.mcgill.ca, gmail.com, mail.mcgill.ca, cs.mcgill.ca, cs.mcgill.ca}

1 Appendix

1.1 Implementation details of the baseline models

Setup. We implemented all the models using the encoder-decoder architecture. The encoders are different baseline models (listed below).The encoder takes as input the given story (paragraph) $p = (p_1, p_2, ...)$ and produces the representation of the story. In all the models, the decoder is implemented as a 2-layer MLP which takes as input the concatenated representation of the story and the embedding of the entities (for which the relationship is to be predicted) and returns a softmax distribution over the relation types. We now describe the different baseline models (encoders) in detail:

LSTM (Hochreiter and Schmidhuber, 1997): The input paragraph is processed by a two-layer Bidirectional LSTM and the hidden state corresponding to the last time-step is used as the representation of the story.

LSTM+attention (Cho et al., 2014): Similar to LSTM, but instead of using just the hidden state at the last timestep, the model computes the attention-weighted mean of the hidden state at all time steps to use as the representation of the story.

Relation Networks - RN (Santoro et al., 2017): An relation module (implemented as an MLP) is used alongside the LSTM to learn pairwise relations among all the pairs of sentences. These relation representations are the output of the relational module. Our input data is prepared as a batch of $sentences \times words$. Each sentence is fed to the LSTM, followed by a pooling (e.g. mean, max) over all hidden states of each sentence to generate the sentence embeddings. The query embeddings are no longer needed in the decoder since they have been incorporated by the relational module when learning relations between sentences.

MAC(Compositional Attention Network) (Hudson and Manning, 2018): A MAC cell is similar to RN, but it also which contains a *control* state c and *memory* state m and can iterate over the input several times. The number of iterations is a hyperparameter. Just like RN, MAC is added behind the LSTM. In each iteration, the model attends to the embeddings of the query entities to generate the current control c_i. Another attention head over c_i and all hidden outputs of LSTM is used to distill the new information r_i. In the end, a linear layer is used to generate the new memory m_i by combining r_i and m_{i-1}. The final memory state gives the representation of the story. This model is the state-of-the-art model for the CLEVER task.

BERT (Devlin et al., 2018): We adapt BERT pre-trained language model to our task. Specifically, we use two variants of BERT - the vanilla 12-layered frozen BERT with pre-trained embeddings, and BERT-LSTM, where a one-layer LSTM encoder is added on top of pretrained BERT embeddings. BERT encodes the sentences into 768-dimensional vectors. To ensure that BERT does not treat the entities as unknown tokens (and hence producing the same representation for all of them), we represent the entities with numbers in the vanilla BERT setup. In BERT-LSTM, we replace the entity embeddings by our entity embedding lookup policy (Refer Appendix 1.5). In both the cases, we use a simple two-layer MLP decoder which takes as inputs the pooled document representation and query representations and produces the softmax distribution over the relations.

Graph Attention Network(GAT) (Veličković et al., 2018): Entity(modelled as nodes in the graph) representations are learned by using the GAT Graph Neural Network with attention-based aggregation over the neighbor nodes. We modify the GAT architecture by attending over each node v_j in the neighborhood of v_i by concatenating the edge

representation $e_{i,j}$ to the representation of v_i.

Relational Recurrent Network (RMC) (Santoro et al., 2018): We also implemented RMC, a recently proposed model for relational reasoning. It works like an RNN, processing words step-by-step, except that a memory matrix ($num\_slots \times mem\_size$) is added as the hidden state. The relational bias is extracted in each step by using self-attention over the concatenation of memory matrix and the word input in the current step. The final memory matrix is the representation of the story. Our implementation is based on another open source implementation. [1]. We noticed that the performance of the model is significantly less across all the tasks by a large margin. Till the time of the submission, we could not verify whether this subpar performance is due to buggy implementation of the code or due to some unexplored hyperparameter combination. Hence we decided not to include the results corresponding to this model in the empirical evaluation. We will continue working on verifying the implementation of the model.

[1] https://github.com/L0SG/relational-rnn-pytorch

1.2 Relations and KB used in CLUTRR Benchmark

$$
\begin{aligned}
[\texttt{grand}, X, Y] &\vdash [[\texttt{child}, X, Z], [\texttt{child}, Z, Y]],\\
[\texttt{grand}, X, Y] &\vdash [[\texttt{SO}, X, Z], [\texttt{grand}, Z, Y]],\\
[\texttt{grand}, X, Y] &\vdash [[\texttt{grand}, X, Z],\\
&\qquad [\texttt{sibling}, Z, Y]],\\
[\texttt{inv-grand}, X, Y] &\vdash [[\texttt{inv-child}, X, Z],\\
&\qquad [\texttt{inv-child}, Z, Y]],\\
[\texttt{inv-grand}, X, Y] &\vdash [[\texttt{sibling}, X, Z],\\
&\qquad [\texttt{inv-grand}, Z, Y]],\\
[\texttt{child}, X, Y] &\vdash [[\texttt{child}, X, Z],\\
&\qquad [\texttt{sibling}, Z, Y]],\\
[\texttt{child}, X, Y] &\vdash [[\texttt{SO}, X, Z],\\
&\qquad [\texttt{child}, Z, Y]],\\
[\texttt{inv-child}, X, Y] &\vdash [[\texttt{sibling}, X, Z],\\
&\qquad [\texttt{inv-child}, Z, Y]],\\
[\texttt{inv-child}, X, Y] &\vdash [[\texttt{child}, X, Z],\\
&\qquad [\texttt{inv-grand}, Z, Y]],\\
[\texttt{sibling}, X, Y] &\vdash [[\texttt{child}, X, Z],\\
&\qquad [\texttt{inv-un}, Z, Y]],\\
[\texttt{sibling}, X, Y] &\vdash [[\texttt{inv-child}, X, Z],\\
&\qquad [\texttt{child}, Z, Y]],\\
[\texttt{sibling}, X, Y] &\vdash [[\texttt{sibling}, X, Z],\\
&\qquad [\texttt{sibling}, Z, Y]],\\
[\texttt{in-law}, X, Y] &\vdash [[\texttt{child}, X, Z],\\
&\qquad [\texttt{SO}, Z, Y]],\\
[\texttt{inv-in-law}, X, Y] &\vdash [[\texttt{SO}, X, Z],\\
&\qquad [\texttt{inv-child}, Z, Y]],\\
[\texttt{un}, X, Y] &\vdash [[\texttt{sibling}, X, Z],\\
&\qquad [\texttt{child}, Z, Y]],\\
[\texttt{inv-un}, X, Y] &\vdash [[\texttt{inv-child}, X, Z],\\
&\qquad [\texttt{sibling}, Z, Y]],
\end{aligned}
$$

In the CLUTRR Benchmark, the following kinship relations are used: *son, father, husband, brother, grandson, grandfather, son-in-law, father-in-law, brother-in-law, uncle, nephew, daughter, mother, wife, sister, granddaughter, grandmother, daughter-in-law, mother-in-law, sister-in-law, aunt, niece*.

We used a small, tractable, and logically sound KB of rules as mentioned above. We carefully select this set of deterministic rules to avoid ambiguity in the resolution. We use gender-neutral predicates and resolve the gender of the predicate in the head $\mathcal{H}$ of a clause $\mathcal{C}$ by deducing the gender of the second constant. We have two types of predicates, *vertical* predicates (parent-child relations) and *horizontal* predicates (sibling or significant other). We denote all the vertical predicates by its *child-to-parent* relation and append the prefix `inv-` to the predicates for the corresponding *parent-to-child* relation. For exam-

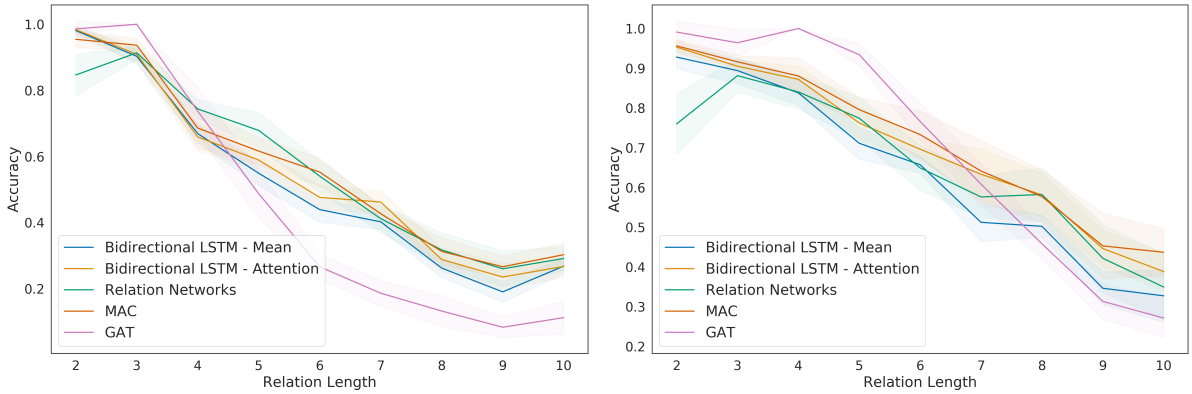

Figure 1: Systematic Generalizability of different models on `CLUTRR-Gen` task (having 20% less placeholders and without training and testing placeholder split), when **Left:** trained with $k = 2$ and $k = 3$ and **Right:** trained with $k = 2, 3$ and 4

ple, `grandfatherOf` is denoted by the gender-neutral predicate $[\texttt{inv-grand}, X, Y]$, where the gender is determined by the gender of Y.

1.3 Effect of placeholder size and split

To analyze whether the language models fail to learn a robust mapping from natural language narratives to underlying logical facts, we re-run the generalization experiments with a reduced placeholder size (20% of the full collected placeholders) *and* we keep the same placeholders for both training and testing. We observe all language-based models are now competitive with respect to GAT on both training regimes $k = 2, 3$ and $k = 2, 3, 4$. This shows the need for separating the placeholder split to effectively test systematic generalization because otherwise, current NLU systems tend to exploit the underlying language layer to arrive at the correct answer.

1.4 More evaluations on Robust Reasoning

We performed several additional experiments to analyze the effect of different training regimes in the Robust Reasoning setup (Table 1) of CLUTRR. Specifically, we want to analyze the effect on zero-shot generalization and robustness when training with different noisy data settings. We notice that the GAT model, having access to the true underlying graph of the puzzles, perform better across different testing scenarios when trained with the noisy data. As the *Supporting facts* contains cycles, it is difficult for GAT to generalize for a dataset with cycles when it is trained on a dataset without cycles. However, when trained with cycles, GAT learns to attend to *all* the paths leading to the correct answer. This effect is disastrous when GAT is tested on *Irrelevant facts* which contains dangling paths as GAT still tries to attend to all the paths. Training on *Irrelevant facts* proved to be most beneficial to GAT, as the model now perfectly attends to *only relevant paths*.

Since *Disconnected facts* contains disconnected paths, the message passing function of the graph is unable to forward any information from the disjoint cliques, thereby having superior testing scores throughout several scenarios.

Experiments on synthetic placeholders. In order to further understand the effect of language placeholders on robustness, we performed another set of experiments where we use bABI (Weston et al., 2015) style simple placeholders (Table 2). We observe a marked increase in performance of all NLU models, where they significantly decrease the gap between their performance with that of GAT, even outperforming GAT on various settings. This shows the significance of using paraphrased placeholders in devising the complexity of the dataset.

1.5 Comparison among different entity embedding policies

In Cloze style reading comprehension tasks, it is sometimes customary to choose UNK embeddings for entity placeholders. (Chen et al., 2016) In our task, however, choosing UNK embeddings for entities is not feasible as the query involves two entities themselves. During preprocessing of our dataset, we convert the entity names into a Cloze-style setup where each entity is replaced by *@entity-n* token. However, one has to be careful not to assign tokens in the same order for all the stories, which will lead to obvious overfitting since the models will learn to work around positional markers as shown

	Models	Unstructured models (no graph)						Structured model (with graph)
Training	Testing	BiLSTM - Attention	BiLSTM - Mean	RN	MAC	BERT	BERT-LSTM	GAT
Supporting	Clean	0.38 ±0.04	0.32 ±0.04	0.4 ±0.09	0.45 ±0.03	0.19 ±0.06	0.39 ±0.06	**0.92** ±0.17
	Supporting	0.67 ±0.06	0.66 ±0.07	0.68 ±0.05	0.65 ±0.04	0.32 ±0.09	0.57 ±0.04	**0.98** ±0.01
	Irrelevant	0.44 ±0.03	0.39 ±0.03	**0.51** ±0.08	0.46 ±0.09	0.2 ±0.06	0.36 ±0.05	0.5 ±0.23
	Disconnected	0.31 ±0.21	0.25 ±0.16	0.47 ±0.08	0.41 ±0.06	0.2 ±0.08	0.32 ±0.04	**0.92** ±0.05
Irrelevant	Clean	0.57 ±0.05	0.56 ±0.05	0.46 ±0.13	0.67 ±0.05	0.24 ±0.06	0.46 ±0.08	**0.92** ±0.0
	Supporting	0.38 ±0.22	0.31 ±0.16	0.61 ±0.07	0.61 ±0.04	0.27 ±0.06	0.46 ±0.04	**0.77** ±0.12
	Irrelevant	0.51 ±0.06	0.52 ±0.06	0.5 ±0.04	0.56 ±0.04	0.25 ±0.06	0.53 ±0.06	**0.93** ±0.01
	Disconnected	0.44 ±0.26	0.54 ±0.27	0.55 ±0.05	0.61 ±0.06	0.26 ±0.03	0.45 ±0.08	**0.85** ±0.25
Disconnected	Clean	0.45 ±0.02	0.47 ±0.03	0.53 ±0.09	0.5 ±0.06	0.22 ±0.09	0.44 ±0.05	**0.75** ±0.07
	Supporting	0.47 ±0.03	0.46 ±0.05	0.54 ±0.03	0.58 ±0.06	0.22 ±0.06	0.38 ±0.08	**0.78** ±0.12
	Irrelevant	0.47 ±0.05	0.48 ±0.03	0.52 ±0.04	0.51 ±0.05	0.17 ±0.04	0.38 ±0.05	**0.56** ±0.26
	Disconnected	0.57 ±0.07	0.57 ±0.06	0.45 ±0.11	0.4 ±0.1	0.17 ±0.05	0.47 ±0.06	**0.96** ±0.01
Average		0.47 ±0.08	0.46 ±0.08	0.52 ±0.07	**0.53** ±0.06	0.23 ±0.07	0.43 ±0.05	**0.82** ±0.09

Table 1: Testing the robustness of the various models when trained various types of noisy facts and evaluated on other noisy / clean facts. The types of noise facts (supporting, irrelevant and disconnected) are defined in Section 3.5 of the main paper.

	Models	Unstructured models (no graph)						Structured model (with graph)
Training	Testing	BiLSTM - Attention	BiLSTM - Mean	RN	MAC	BERT	BERT-LSTM	GAT
Supporting	Clean	0.96 ±0.01	**0.97** ±0.01	0.88 ±0.05	0.94 ±0.02	0.48 ±0.08	0.57 ±0.08	0.92 ±0.17
	Supporting	0.96 ±0.03	0.96 ±0.03	0.97 ±0.01	0.97 ±0.01	0.75 ±0.07	0.88 ±0.05	**0.98** ±0.01
	Irrelevant	0.92 ±0.02	**0.93** ±0.01	0.9 ±0.03	0.91 ±0.01	0.56 ±0.04	0.54 ±0.06	0.5 ±0.23
	Disconnected	0.8 ±0.04	0.83 ±0.04	0.76 ±0.08	0.86 ±0.04	0.27 ±0.06	0.42 ±0.08	**0.92** ±0.05
Irrelevant	Clean	0.63 ±0.02	0.61 ±0.07	0.85 ±0.09	0.8 ±0.07	0.53 ±0.09	0.44 ±0.06	**0.92** ±0.0
	Supporting	0.66 ±0.03	0.64 ±0.04	0.69 ±0.06	0.76 ±0.06	0.42 ±0.08	0.43 ±0.08	**0.77** ±0.12
	Irrelevant	0.89 ±0.04	0.86 ±0.1	0.74 ±0.11	0.78 ±0.06	0.61 ±0.1	0.83 ±0.06	**0.93** ±0.01
	Disconnected	0.64 ±0.02	0.62 ±0.05	0.72 ±0.05	0.73 ±0.04	0.41 ±0.04	0.61 ±0.05	**0.85** ±0.25
Disconnected	Clean	0.9 ±0.05	0.82 ±0.12	**0.94** ±0.02	0.93 ±0.04	0.68 ±0.07	0.64 ±0.02	0.75 ±0.07
	Supporting	0.87 ±0.04	0.82 ±0.05	0.85 ±0.03	**0.88** ±0.04	0.54 ±0.08	0.5 ±0.05	0.78 ±0.12
	Irrelevant	**0.87** ±0.03	0.85 ±0.03	0.83 ±0.03	0.87 ±0.02	0.59 ±0.09	0.58 ±0.09	0.56 ±0.26
	Disconnected	0.91 ±0.04	0.91 ±0.03	0.8 ±0.17	0.71 ±0.11	0.49 ±0.1	0.79 ±0.1	**0.96** ±0.01
Average		0.83 ±0.08	0.82 ±0.08	0.83 ±0.07	**0.84** ±0.06	0.58 ±0.07	0.60 ±0.05	**0.82** ±0.09

Table 2: Testing the robustness on toy placeholders of the various models when trained various types of noisy facts and evaluated on other noisy / clean facts. The types of noise facts (supporting, irrelevant and disconnected) are defined in Section 3.5 of the main paper.

in Chen et al. (2016). Therefore, we randomize the Cloze-style entities themselves for each story. We experimented with three different policies of choosing the entity embeddings:

1. *Fixed Random Embeddings*: One simple and intuitive choice is to assign a random embedding to each entity and keep it fixed throughout the training. During our data-processing pipeline, we ensure that all the entity tokens are randomized using a pool of entity tokens, hence the chances of a model learning to exploit the positional markers are slim.

2. *Randomized Random Embeddings*: We can go one step further and randomize the random embeddings at *each epoch*. This aggressive strategy does not let the model learn any positional markings at all, however it might hamper the learning ability of models as the entity representations are changing arbitrarily.

3. *Learned Random Embeddings*: Since our data pre-processing pipeline randomly assigns the entities on each story, we can as well learn a pool of n entities, from which a subset is always used to replace the entities.

We chose to report all experiments with respect to fixed random embeddings. We compared different embedding policies with respect to the Systematic Generalization task. We show a comparison between the Bidirectional LSTM and GAT in Figure 2. We see that the fixed embedding policy has better Systematic Generalization score, although the advantage is minor compared to the other schemes. For GAT, the advantage is practically nil for the different schemes which shows that a Graph Neural Network performs inductive reasoning in the same manner irrespective of the initial node embedding representation.

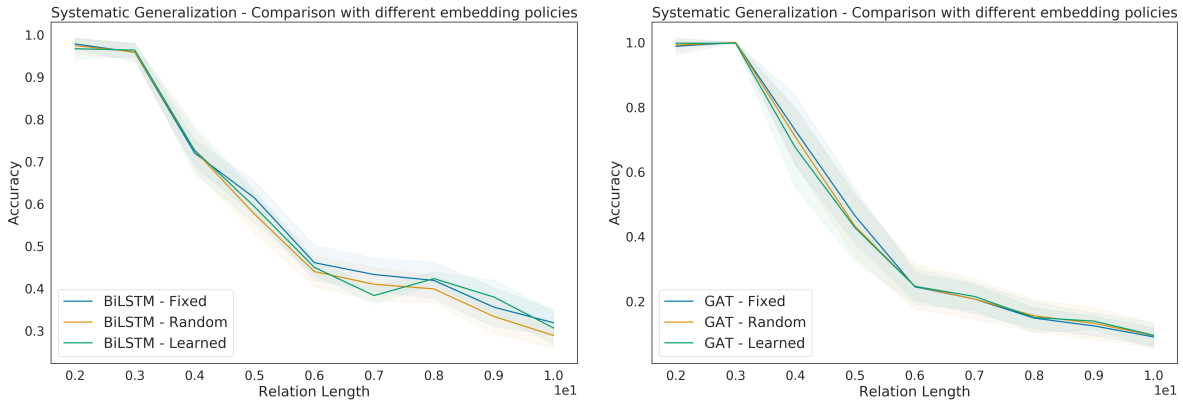

Figure 2: Systematic Generalization comparison with different Embedding policies

1.6 AMT Data collection process

We use ParlAI (Miller et al., 2017) Mturk interface to collect paraphrases from the users. Specifically, given a set of facts, we ask the users to paraphrase the facts into a story. The users (*turkers*) are free to construct any story they like as long as they mention all the entities and all the relations among them. We also provide the head $\mathcal{H}$ of the clause as an *inferred* relation and specifically instruct the users to *not* mention it in the paraphrased story. In order to evaluate the paraphrased stories, we ask the turkers to peer review a story paraphrased by a different turker. Since there are two tasks - paraphrasing a story and rating a story - we choose to pay 0.5$ for each annotation. A sample task description in our MTurk interface is as follows:

> In this task, you will need to write a short, simple story based on a few facts. **It is crucial that the story mentions each of the given facts at least once.** The story does not need to be complicated! It just needs to be grammatical and mention the required facts.
>
> After writing the story, you will be asked to evaluate the quality of a generated story (based on a different set of facts). **It is crucial that you check whether the generated story mentions each of the required facts.**
>
> *Example of good and bad stories: Good Example*
>
> **Facts to Mention**
>
> - John is the father of Sylvia.
> - Sylvia has a brother Patrick.
>
> **Implied Fact**: John is the father of Patrick.
>
> **Written story**
>
> John is the proud father of the lovely Sylvia. Sylvia has a love-hate relationship with her brother Patrick.
>
> *Bad Example*
>
> **Facts to Mention**
>
> - Vincent is the son of Tim.
> - Martha is the wife of Tim.
>
> **Implied Fact** : Martha is Vincent's mother.
>
> **Written story**
>
> Vincent is married at Tim and his mother is Martha.
>
> *The reason the above story is bad*:
>
> - This story is bad because it is nonsense / ungrammatical.
> - This story is bad because it does not mention the proper facts.
> - This story is bad because it reveals the implied fact.

To ensure that the turkers are providing high-quality annotations without revealing the inferred fact, we also launch another task to ask the turkers to rate three annotations to be either good or bad which are provided by a set of *different* turkers. We pay 0.2$ for each HIT consisting of three reviews. This helped to remove logical and grammatical inconsistencies to a large extent. Based on the reviews, 79% of the collected paraphrases passed the peer-review sanity check where all the reviewers agree on the quality. This subset of the placeholders is used in the benchmark. A sample of programmatically generated dataset for clause length of $k = 2$ to $k = 6$ is provided in the tables 5 to 9.

1.7 Human Evaluation

We performed a human evaluation study to analyze the difficulty of our proposed benchmark suite, which is provided in Table 3. We perform the evaluation in two scenarios: first a time-limited scenario where we ask AMT Turkers to solve the puzzle in a fixed time. Turkers were provided a maximum time of 30 mins, but they solved the puzzles in an average of 1 minute 23 seconds. Secondly, we use

Relation Length	Human Performance		Reported Difficulty
	Time Limited	Unlimited Time	
2	0.848	1	1.488 +- 1.25
3	0.773	1	2.41 +- 1.33
4	0.477	1	3.81 +- 1.46
5	0.424	1	3.78 +- 0.96
6	0.406	1	4.46 +- 0.87

Table 3: Human performance accuracies on CLUTRR dataset. Humans are provided the Clean-Generalization version of the dataset, and we test on two scenarios: when a human is given limited time to solve the task, and when a human is given unlimited time to solve the task. Regardless of time, our evaluators provide a score of difficulty of individual puzzles.

another set of expert evaluators who are given ample time to solve the tasks. Not surprisingly, if a human being is given ample time (experts took an average of 6 minutes per puzzle) and a pen and a paper to aid in the reasoning, they get all the relations correct. However, if an evaluator is short of time, they might miss important details on the relations and perform poorly. Thus, our tasks require *active attention*.

In both cases, we asked Turkers and our expert human evaluators to rate the difficulty of a given task in a Likert scale of 1-5, where 1 corresponds to very easy and 5 corresponds to very hard perceived difficulty. This score increases as we increase the complexity of the task by increasing the relations, thereby suggesting that a human being perceives similar difficulty while solving for larger relation tasks. However, since a human being is a systematic learner, given enough time they can solve all puzzles with perfect accuracy. We set aside the task of testing noisy scenarios of CLUTRR to human evaluators as future work.

References

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2358–2367.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

Drew Arad Hudson and Christopher D. Manning. 2018. Compositional attention networks for machine reasoning. In *International Conference on Learning Representations*.

Alexander Miller, Will Feng, Dhruv Batra, Antoine Bordes, Adam Fisch, Jiasen Lu, Devi Parikh, and Jason Weston. 2017. Parlai: A dialog research software platform. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 79–84.

Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. 2018. Relational recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 7310–7321.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards AI-Complete question answering: A set of prerequisite toy tasks.

2 Supplemental Material

To promote reproducibility, we follow the guidelines proposed by the Machine Learning Reproducibility Checklist [2] and release the following information regarding the experiments conducted by our benchmark suite.

2.1 Details of datasets used

A downloadable link to the datasets used can be found here [3]. Details of the individual datasets can be found in Table 4. For all experiments, we use 10,000 training examples and a 100 testing example for each testing scenario. We split the training data 80-20 into a dev set randomly on each run.

[2] Machine Learning Reproducibility Checklist

[3] Dataset link in Google Drive

Dataset	Variant - Training	Variant - Testing	Training Clause length	Testing Clause length
data_089907f8 data_db9b8f04	Clean - Generalization	Clean - Generalization	$(k = 2, 3)$ $(k = 2, 3, 4)$	$(k = 2, 3, \ldots, 10)$
data_7c5b0e70	Clean	Clean, Supporting, Irrelevant, Disconnected	$(k = 2, 3)$	$(k = 2, 3)$
data_06b8f2a1	Supporting	Clean, Supporting, Irrelevant, Disconnected	$(k = 2, 3)$	$(k = 2, 3)$
data_523348e6	Irrelevant	Clean, Supporting, Irrelevant, Disconnected	$(k = 2, 3)$	$(k = 2, 3)$
data_d83ecc3e	Disconnected	Clean, Supporting, Irrelevant, Disconnected	$(k = 2, 3)$	$(k = 2, 3)$

Table 4: Details of publicly released data

2.2 Details of Hyperparameters used

For all models, the common hyperparameters used are: Embedding dimension: 100 (except BERT based models), Optimizer: Adam, Learning rate: 0.001, Number of epochs: 100, Number of runs: 10. Specific model-based hyperparameters are given as follows:

- **Bidirectional LSTM**: LSTM hidden dimension: 100, # layers: 2, Classifier MLP hidden dimension: 200
- **Relation Networks**: f_{θ_1} : 256, f_{θ_2}: 64, g_θ : 64
- **MAC**: # Iterations: 6, `shareQuestion`: True, Dropout - Memory, Read and Write: 0.2
- **Relational Recurrent Networks**: Memory slots: 2, Head size: 192, Number of heads: 4, Number of blocks : 1, forget bias : 1, input bias: 0, gate style: unit, key size: 64, # Attention layers: 3, Dropout: 0
- **BERT**: Layers : 12, Fixed pretrained embeddings from `bert-base-uncased` using Pytorch HuggingFace BERT repository [4], Word dimension: 768, appended with a two-layer MLP for final prediction.
- **BERT-LSTM**: Same parameters as above, with a two-layer unidirectional LSTM encoder on top of BERT word embeddings.
- **GAT**: Node dimension: 100, Message dimension: 100, Edge dimension: 20, number of rounds: 3

[4] https://github.com/huggingface/pytorch-pretrained-BERT

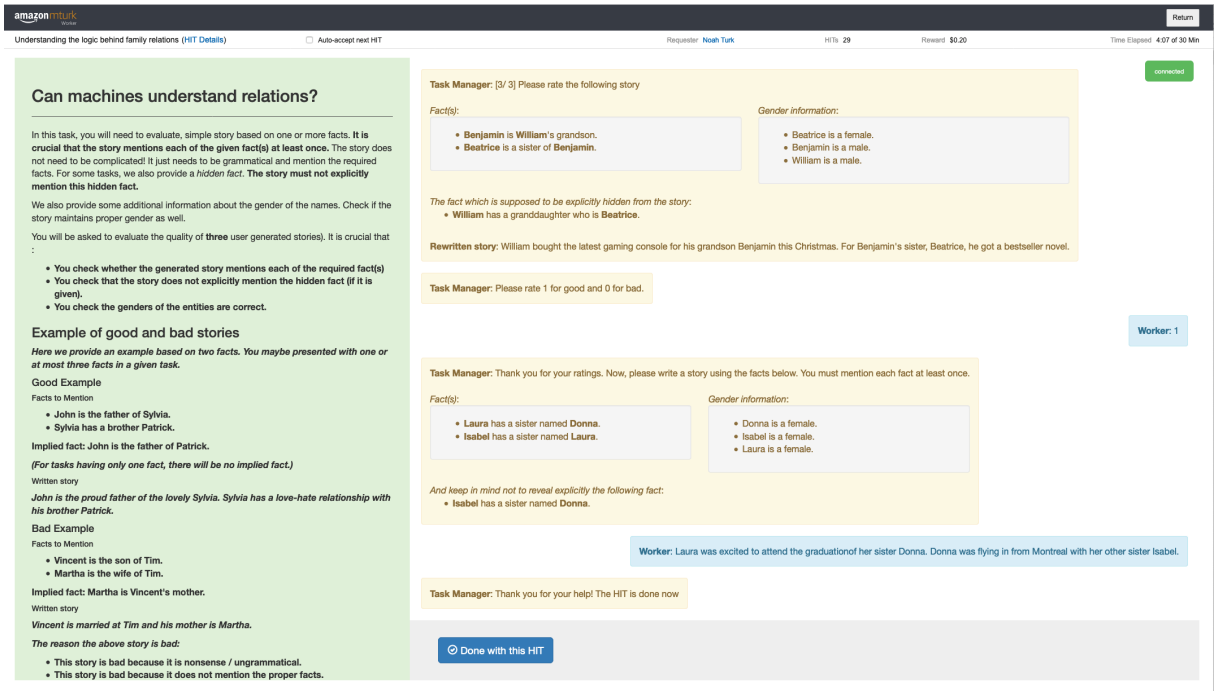

Figure 3: Amazon Mechanical Turker Interface built using ParlAI which was used to collect data as well as peer reviews.

Table 5: Snapshot of puzzles in the dataset for k=2

Puzzle	Question	Gender	Answer
Charles's son *Christopher* entered rehab for the ninth time at the age of thirty. *Randolph* had a nephew called *Christopher* who had n't seen for a number of years.	Randolph is the _____ of Charles	Charles:male, Christopher:male, Randolph:male	brother
Randolph and his sister *Sharon* went to the park. *Arthur* went to the baseball game with. his son *Randolph*	Sharon is the _____ of Arthur	Arthur:male, Randolph:male, Sharon:female	daughter
Frank went to the park with his father, *Brett*. *Frank* called his brother *Boyd* on the phone. He wanted to go out for some beers.	Brett is the _____ of Boyd	Boyd:male, Frank:male, Brett:male	father

Table 6: Snapshot of puzzles in the dataset for k=3

Puzzle	Question	Gender	Answer
Roger was playing baseball with his sons *Sam* and *Leon*. *Sam* had to take a break though because he needed to call his sister *Robin*.	Leon is the _____ of Robin	Robin:female, Sam:male, Roger:male, Leon:male	brother
Elvira and her daughter *Nancy* went shopping together last Monday and they bought new shoes for *Elvira*'s kids. *Pedro* and his sister *Allison* went to the fair. *Pedro*'s mother, *Nancy*, was out with friends for the day.	Elvira is the _____ of Allison	Allison:female, Pedro:male, Nancy:female, Elvira:female	grandmother
Roger met up with his sister *Nancy* and her daughter *Cynthia* at the mall to go shopping together. *Cynthia*'s brother *Pedro* was going to be the star in the new show.	Pedro is the _____ of Roger	Roger:male, Nancy:female, Cynthia:female, Pedro:male	nephew

Table 7: Snapshot of puzzles in the dataset for k=4

Puzzle	Question	Gender	Answer
Celina has been visiting her sister, *Fran* all week. *Fran* is also the daughter of *Bethany*. *Ronald* loves visiting his aunt *Bethany* over the weekends. *Samuel*'s son *Ronald* entered rehab for the ninth time at the age of thirty.	Celina is the _____ of Samuel	Samuel:male, Ronald:male, Bethany:female, Fran:female, Celina:female	niece
Celina adores her daughter *Bethany*. *Bethany* loves her very much, too. *Jackie* called her mother *Bethany* to let her know she will be back home soon. *Thomas* was helping his daughter *Fran* with her homework at home. Afterwards, *Fran* and her sister *Jackie* played Xbox together.	Celina is the _____ of Thomas	Thomas:male, Fran:female, Jackie:female, Bethany:female, Celina:female	daughter
Raquel is *Samuel* 'daughter and they go shopping at least twice a week together. *Kenneth* and her mom, *Theresa*, had a big fight. *Theresa*'s son, *Ronald*, refused to get involved. *Ronald* was having an argument with her sister, *Raquel*.	Samuel is the _____ of Kenneth	Kenneth:male, Theresa:female, Ronald:male, Raquel:female, Samuel:male	father

Table 8: Snapshot of puzzles in the dataset for k=5

Puzzle	Question	Gender	Answer
Steven's son is *Bradford*. *Bradford* and his father always go fishing together on Sundays and have a great time together. *Diane* is taking her brother *Brad* out for a late dinner. *Kristin*, *Brad*'s mother, is home with a cold. *Diane*'s father *Elmer*, and his brother *Steven*, all got into the rental car to start the long cross-country roadtrip they had been planning.	Bradford is the _____ of Kristin	Kristin:female, Brad:male, Diane:female, Elmer:male, Steven:male, Bradford:male	nephew
Elmer went on a roadtrip with his youngest child, *Brad*. *Lena* and her sister *Diane* are going to a restaurant for lunch. *Lena*'s brother *Brad* is going to meet them there with his father *Elmer Brad* ca n't stand his unfriendly aunt *Lizzie*.	Lizzie is the _____ of Diane	Diane:female, Lena:female, Brad:male, Elmer:male, Lizzie:female	aunt
Ira took his niece *April* fishing Saturday. They caught a couple small fish. *Ronald* was enjoying spending time with his parents, *Damion* and *Claudine*. *Damion*'s other son, *Dennis*, wanted to come visit too. *Dennis* often goes out for lunch with his sister, *April*.	Ira is the _____ of Claudine	Claudine:female, Ronald:male, Damion:male, Dennis:male, April:female, Ira:male	brother

Table 9: Snapshot of puzzles in the dataset for k=6

Puzzle	Question	Gender	Answer
Mario wanted to get a good gift for his sister, *Marianne*. *Jean* and her sister *Darlene* were going to a party held by *Jean*'s mom, *Marianne*. *Darlene* invited her brother *Roy* to come, too, but he was too busy. *Teri* and her father, *Mario*, had an argument over the weekend. However, they made up by Monday. *Agnes* wants to make a special meal for her daughter *Teri*'s birthday.	Roy is the _____ of Agnes	Agnes:female, Teri:female, Mario:male, Marianne:female, Jean:female, Darlene:female, Roy:male	nephew
Robert's aunt, *Marianne*, asked *Robert* to mow the lawn for her. *Robert* said he could n't because he had a bad back. *William*'s parents, *Brian* and *Marianne*, threw him a surprise party for his birthday. *Brian*'s daughter *Jean* made a mental note to be out of town for her birthday! *Agnes*'s biggest accomplishment is raising her son *Robert*. *Jean* is looking for a good gift for her sister *Darlene*.	Darlene is the _____ of Agnes	Agnes:female, Robert:male, Marianne:female, William:male, Brian:male, Jean:female, Darlene:female	niece
Sharon and her brother *Mario* went shopping. *Teri*, *Mario*'s daughter, came too. *Agnes*, *Annie*'s mother, is unhappy with *Robert*. She feels her son is cruel to *Annie*'s sister *Teri*, and she wants *Robert* to be nicer. *Robert*'s sister, *Nicole*, participated in the dance contest.	Nicole is the _____ of Sharon	Sharon:female, Mario:male, Teri:female, Annie:female, Agnes:female, Robert:male, Nicole:female	niece