Few Shot Adaptation to New Grammars in Instruction Based Learning

Kanishk Gandhi
New York University
New York, NY 10023
kvg245@nyu.edu

Urvish Desai
New York University
New York, NY 10023
urvish@nyu.edu

Abstract

Designing agents that learn from natural language instructions and interact with humans in the form of language to modify or enhance their behaviour is highly desirable. Further, agents that can understand previously unseen grammars with a few examples is advantageous. Here we propose using an environment called BabyAI to train and observe learning in reinforcement learning and imitation learning agents. We explore training an imitation learning agent with meta learning, specifically Model Agnostic Meta Learning and try to see how well it can generalize to previously unseen grammars.

1 Introduction

Understanding natural language instructions is one of the highest cognitive capabilities of humans. Cognitive psychologists have long tried to study these aspects of human learning in children. Learning from natural language is a fundamental task for an AI. Imagining an intelligent machine learning a new task from language or modifying how it does an existing task is natural. Generalizing to a novel set of actions should also be trivial for an optimal learner. Humans usually learn language with visual cues (Lake et al. [2017]) and it should be interesting to explore the relations developed between the visual and the lingual domain.

The data hungry nature of current Neural Language models makes it difficult for them to generalize quickly to a new syntax of the language. There are several situations where it is desirable for an agent to generalize to a new language or a new grammar, with somewhat different syntax and semantics. We emulate this situation by generating different grammars for the existing instructions and try to see if an agent can generalize to them. We test how well the agent can learn to understand new grammars with as low as 10 examples with a small number of gradient updates.

Traditionally, machine learning involves accumulating a huge dataset on a very specific problem and then leveraging this data to train models for that specific task. In contrast, humans do not require as much data and can generalize with a very few examples. As shown in the figure from Lake et al. [2015], humans can immediately recognize varying instances of objects from a single example. This is because of our capacity to learn to learn.

Learning to learn a new concept has been associated with one of the most difficult cognitive abilities to replicate in artificial networks. This has prompted the research community to work towards trying to design better paradigms that learn quickly. Initial attempts at learning to learn or meta-learning can be traced to Jurgen Schmiduber's Schmidhuber [1987] work and the work by the Bengio brothers (Bengio et al. [1990]). With Lake et al. [2015] posing a challenge to the community to design algorithms that learn new concepts quickly, there has been a burst of papers in the domain of meta-learning which particularly focus on hyperparameter optimization using neural networks for image recognition in a small number of examples and quick RL.

Figure 1: Humans can learn new classes and associated concepts through samll amounts of data (Lake et al. [2015])

We choose to use a meta-learning paradigm to learn a new grammar for a variety of tasks using the popular Model-Agnostic Machine LearningFinn et al. [2017] paradigm.

In this report, we experiment with an appropriately named toy environment called BabyAI (Chevalier-Boisvert et al. [2018]), that provides simple tasks based on generated instructions that an agent must learn to follow. In subsequent sections, we look at how we generate the data, handle the meta learning process and then look at how our model performs relative to a few baselines. As we use a model agnostic training regime, we keep the models used in Chevalier-Boisvert et al. [2018] the same. We further show how one could explore meta learning at the scale of tasks to reduce examples required for training and also extend it to reinforcement learning.

The idea behind generalizing the grammar is to get a better insight into the compositional structure of the language. This could also improve the accuracy of a pre-trained language model.

2 Related Work

Recurrent Neural Networks like GRUs (Cho et al. [2014]) and LSTMs Hochreiter and Schmidhuber [1997] have been shown to capture the to encode the instructions and a convolutional network with two batch normalized (Ioffe and Szegedy [2015]) FiLM (Perez et al. [2017]) layers. The use of FiLM layers and their application to reinforcement or imitation learning is pretty interesting. FiLM layers can essentially be viewed as a generalized form of conditional normalization layers and have been seen to perform particularly well on a variety of tasks, particularly visual reasoning tasks. The reinforcement learning agent for the baseline is trained using Proximal Policy Optimization (Schulman et al. [2017]) and with paralelized data collection.

Work on reducing the data required for training agents has been an active research problem. Recent solutions leverage simple models in cognitive sciences like interaction networks (Battaglia et al. [2016]) and relation networks (Santoro et al. [2017]) that construct a graph between concepts of a problem and try to model mutual relations and effective using neural networks have shown impressive results.

Work on using a hierarchical model (HIRO) (Nachum et al. [2018]) of goal setting in reinforcement learning has made some progress in recent times. A combination of a model-based and a model-free Reinforcement Learning model, that might predict future trajectories or strategies (I2A) (Weber et al. [2017]) and rewards might be useful in our use-case here. Unsupervised learning as in other domains has been one of the common approach taken. Using auxiliary tasks while training as in UNREAL (Jaderberg et al. [2016]) significantly reduced training time and improved performance.

Using generative models like VAEs (Kingma and Welling [2013]) to capture the compositionality in the environment as was done in DARLA (Higgins et al. [2017]), is found to reduce the data requirement. Using some notion of memory is shown to be useful in MERLIN (Wayne et al. [2018]) to learn to solve tasks quicker as agents are seen to refer to past experiences while taking decisions to improve performance.

(Lake and Baroni [2018]) try exploring the generalization of models in NLP, particularly sequence to sequence models and their shortcomings to generalize to novel combinations of previously seen instances. We try exploring this aspect of understanding models on encoding language in a new setting by the generation of new grammars.

Learning to learn has been popular problem in the research community to create networks that are less data hungry Andrychowicz et al. [2016]. The vision community has made great strides Vinyals et al. [2016] Ravi and Larochelle [2016] by optimizing the initial states of networks on datasets such as the Omniglot Lake et al. [2015] and MiniImagenet Ravi and Larochelle [2016] . Finn et al. [2017] was a landmark paper in this domain, which provided a simple model agnostic method to learn to initialize networks at an optimal stage. We use the technique as described in this paper to train our network. MAML has been further worked upon to make it more general in REPTILE Nichol et al. [2018]. These techniques have subsequently been tried on learning language models that provide simpler unsupervised learning rules. Metz et al. [2018] is a survey in this domain that provides a good overview of the paradigms used.

3 Environment

The BabyAI environment (Chevalier-Boisvert et al. [2018]) is designed specifically with instruction based natural language reinforcement learning and imitation learning. It further provides a bot script that could be used to provide a faux human in the loop kind of setting to help evaluate agent performance.

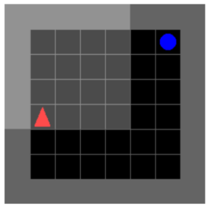
The model includes a set of basic tasks in a 7×7 grid with different levels of difficulty (A staircase of tasks). As the difficulty increases, the size of these grids is also increased. This provides for a way to incorporate curriculum learning which has been seen to benefit efficient learning (Bengio et al. [2009] (Kumar et al. [2010] (Zaremba and Sutskever [2014]). Some tasks are shown in Figure 1.

The Baby language is a restricted language which is used to encode a certain set of goals and actions. The language although being restricted still provides around $2.48 \times 10^{1}9$ combinations of possible sequences. For example, some of the sample Baby instructions are:

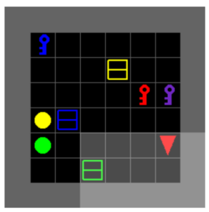
- go to the red ball
- open the door on your left
- put a ball next to the blue door
- open the yellow door and go to the key behind you
- put a ball next to a purple door after you put a blue box next to a grey box and pick up the purple box

We break down the instructions into their basic part of speech tags in order to shuffle them. A context free grammar is used to create different grammars and generate new instructions. For example, we show a few ways of parsing for the first 'GO TO' task and how about 120 grammars could be generated for the first task.

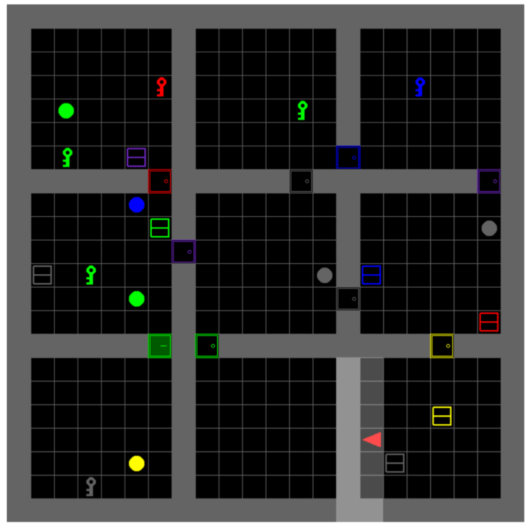
- <VB> <PRT-TO> <DT> <JJ> <NN-JJ> / <VB> <DT> <NN-DT> <IN> <TO> <DT> <JJ> <NN-JJ>
  - go to the red ball
  - put a ball next to a purple door
- <VB> <PRT-TO> <NN-JJ> <DT> <JJ> / <VB> <NN-DT> <DT> <IN> <TO> <NN-JJ> <DT> <JJ>
  - go to ball the red
  - put ball a next to door a purple



(a) GoToObj: "go to the blue ball"



(b) PutNextLocal: "put the blue key next to the green ball"



(c) BossLevel: "pick up the grey box behind you, then go to the grey key and open a door". Note that the green door near the bottom left needs to be unlocked with a green key, but this is not explicitly stated in the instruction.

Figure 2: Some of the Baby AI levels. The red triangle represents the agent, while the light gray region is its field of view. The boss level is the final level in the staircase of tasks (Chevalier-Boisvert et al. [2018])

$$
\begin{array}{rcl}
\langle\text{Sent}\rangle & \models & \langle\text{Sent1}\rangle \;|\; \langle\text{Sent1}\rangle \text{ ',' then } \langle\text{Sent1}\rangle \;|\; \langle\text{Sent1}\rangle \text{ after you } \langle\text{Sent1}\rangle \\
\langle\text{Sent1}\rangle & \models & \langle\text{Clause}\rangle \;|\; \langle\text{Clause}\rangle \text{ and } \langle\text{Clause}\rangle \\
\langle\text{Clause}\rangle & \models & \text{go to } \langle\text{Descr}\rangle \;|\; \text{pick up } \langle\text{DescrNotDoor}\rangle \;|\; \text{open } \langle\text{DescrDoor}\rangle \;| \\
 & & \text{put } \langle\text{DescrNotDoor}\rangle \text{ next to } \langle\text{Descr}\rangle \\
\langle\text{DescrDoor}\rangle & \models & \langle\text{Article}\rangle\ \langle\text{Color}\rangle \text{ door } \langle\text{LocSpec}\rangle \\
\langle\text{DescrBall}\rangle & \models & \langle\text{Article}\rangle\ \langle\text{Color}\rangle \text{ ball } \langle\text{LocSpec}\rangle \\
\langle\text{DescrBox}\rangle & \models & \langle\text{Article}\rangle\ \langle\text{Color}\rangle \text{ box } \langle\text{LocSpec}\rangle \\
\langle\text{DescrKey}\rangle & \models & \langle\text{Article}\rangle\ \langle\text{Color}\rangle \text{ key } \langle\text{LocSpec}\rangle \\
\langle\text{Descr}\rangle & \models & \langle\text{DescrDoor}\rangle \;|\; \langle\text{DescrBall}\rangle \;|\; \langle\text{DescrBox}\rangle \;|\; \langle\text{DescrKey}\rangle \\
\langle\text{DescrNotDoor}\rangle & \models & \langle\text{DescrBall}\rangle \;|\; \langle\text{DescrBox}\rangle \;|\; \langle\text{DescrKey}\rangle \\
\langle\text{LocSpec}\rangle & \models & \epsilon \;|\; \text{on your left} \;|\; \text{on your right} \;|\; \text{in front of you} \;|\; \text{behind you} \\
\langle\text{Color}\rangle & \models & \epsilon \;|\; \text{red} \;|\; \text{green} \;|\; \text{blue} \;|\; \text{purple} \;|\; \text{yellow} \;|\; \text{grey} \\
\langle\text{Article}\rangle & \models & \text{the} \;|\; \text{a}
\end{array}
$$

Figure 3: The Backus-Naur Form grammar for instructions with 2.48 1019 possible combinations (Chevalier-Boisvert et al. [2018])

- <DT> <JJ> <NN-JJ> <VB> <PRT-TO> / <DT> <NN-DT> <VB> <IN> <TO> <DT> <JJ> <NN-JJ>
  - the red ball go to
  - a ball put next to a purple door

4 Evaluation Metrics

The environment considers a task to be successful if the agent reaches a success rate of at least 99%. One of the main motives of this environment was to make the data required for training smaller and in essence make reinforcement learning or imitation learning with a human in the loop more practical. Deviating from the original metrics used, we measure the accuracy of the imitation learning agent with respect to an expert bot to try and measure, how quickly it learns to perform in an optimal fashion. We test the performance on different number of gradient updates during the validation procedure where we fine tune the initial meta-learned network.

5 Experiments

5.1 Model Design

The instruction reading section of the model is a GRU with 128 hidden size and uses attention Bahdanau et al. [2014]Luong et al. [2015]. A FiLM layer is used to conditionally normalize the feature maps from the convolutional layers. This conditional normalization has shown to aid performance in tasks that require visual reasoning. The MAML learning algorithm used is shown in the figure. All code for this project can be found at http://github.com/kanishkg/babyai.

The concept used for meta-learning is simple, but very impactful. The intuition behind the MAML algorithm is to train the neural network parameters over a distribution of tasks and adjust the parameters on encountering a new unseen task with few meta-gradient steps. Thus, it uses meta-learning to learn the parameters of any standard model.

Algorithm 1 MAML Trained over multiple grammars

Require: p(G) Distribution over the grammars
Require: Optimizer hyperparameters
randomly initialize θ
while not done **do**
    $G_i \leftarrow$ Sample batch of Grammars (p(G))
    for all G_i **do**
        Evaluate $\nabla_\theta L_{Gi}(f_\theta)$ with respect to K examples
        Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta L_{Gi}(f_\theta)$
    end for
    Update $\theta \leftarrow \theta - \beta \nabla_\theta \Sigma_{G_i\ p(G)} L_{Gi}(f_{\theta_i'})$
end while

The loss is a combination of entropy of decision and the difference in policy from the expert bot. The weighting factor in the loss for the entropy is set at 0.01. The gradient is propagated backward in time for 20 timesteps. Further, the embedding sizes were as follows:

- Image Embedding : 128
- Memory Dimension : 128
- Instruction Embedding : 128

5.2 Training Procedure

The environment is used in conjunction with an expert bot to create demonstrations for the imitation learning agent to learn from. About 1000 examples are generated for the training set, while 500 are kept for validating the model. The signal for which grammar is being used is given by appending the grammar type to the text.

In this set of experiments, the agent is only trained on the 'go to' task. A set of grammars is generated for this task according to which the model is trained. For each meta-batch, the model is trained on 100 grammars. The remaining grammars are held out for validation. The inner model is updated with a learning rate of 0.4, while the meta learning rate is set at 0.1 with an Adam optimizer. A batch size of 32 demonstrations is used while training.

For the validation, a small batches of size 10 are randomly sampled from the validation set of demos and gradient updates are performed in this set. We perform 1, 5 and 10 gradient updates with a learning rate of 0.4 to see how well the model learns in 10 'shots'.

5.3 Pretrained Model

A pretrained expert imitation learning model, that achieves an accuracy of about 99.9% on the English grammar is fine tuned on a new grammar using 1, 5, 10 gradient steps.

It is expected that this model will give a very strong result considering that the visual features should have been extracted extremely well from the task that the agent was previously trained upon.

5.4 Results

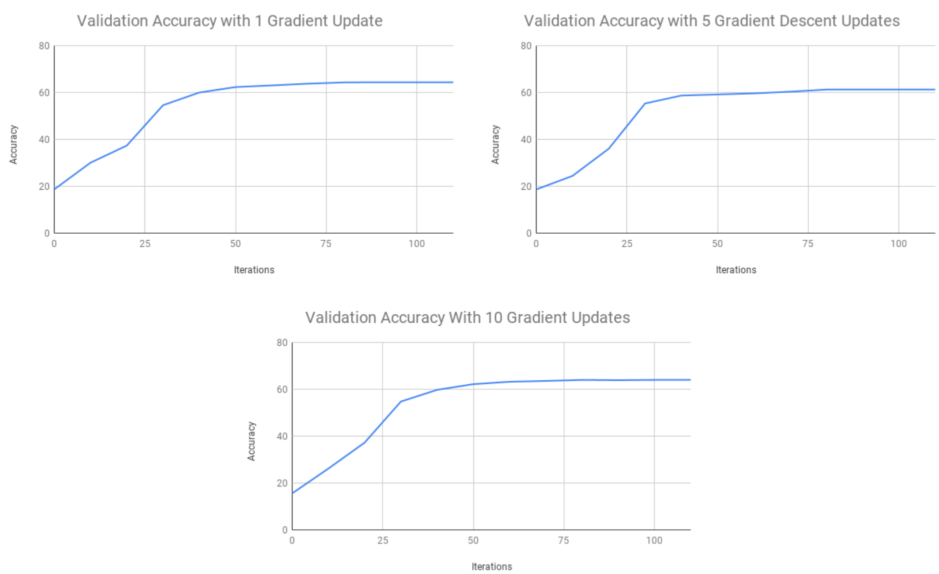


Figure 4: Validation accuracies for varying number of gradient updates

We fine-tuned the meta learned model and the pre-trained model with a learning rate of 0.4 for a batch size of 10 and observed the accuracies on the validation set. The iterations shown in the above figures are for meta updates. It can be seen that the validation accuracies achieved for 1 gradient update and 10 gradient updates are extremely similar, while the accuracy for 5 gradient updates is lower.

Number of Gradient Updates	Pre-Trained	Meta-Learned
1	64.40%	64.38%
5	64.70%	61.25%
10	65.20%	64%

Table 1: Comparison of fine-tuning a pre-trained model and using a model learnt using meta learning for varying number of gradient updates

We would like to say that the experiments using meta-learning weren't successful as the accuracy of the model with this paradigm weren't much higher than fine-tuning a pre-trained model. Thus, we weren't unable to transfer the knowledge from the variation in training grammars to the model parameters. We believe that this could be due to a lack of variation among tasks. Training the model jointly with more tasks in the environment could potentially solve this issue.

6 Conclusion

We see that we can achieve a decent level of accuracy with a very small number of gradient updates on a small set of examples. It can be seen that the initialization of the model is highly optimized. This could be because the network is initialized in such a way that it is optimal to extract the compositional information in the language irrespective of the grammar. We believe that while conventional neural language models do not have syntactic systematicity, a model trained in this fashion with multiple grammars has a sense of syntactic systematicity. It should be able to generalize to new words and grammars efficiently.

Further, we do not see an extremely high performance as the variation in the tasks is only with the lingual domain, while the overall task remains the same. We believe that this is not optimal for learning good initializations for the visual layers and consequently the layers requiring us to conditionally normalize the visual features based on the instruction. This could be the reason that the model does not perform exceedingly well and has a performance just about comparable to the fine-tuned pre-trained model.

7 Discussion and Future Work

It should be noted here that we did not train multiple tasks with the same grammar in conjunction with each other. We expect that training in such a fashion should boost the ability of the meta learning algorithm to initialize the network even better. Similar experiments can be run with an RL agent although the training times would be much longer due to the sparser reward signals.

The original goal of the environment was to reduce the training iterations required to train the agent and we think that a distribution of tasks in line with the tasks predefined in the environment could be used to train the agent. Although the initial set of tasks do not provide enough variation to train a meta-learning algorithm, some new variations of these tasks could be created, or an unsupervised way to generate tasks could be explored. Gupta et al. [2018]. An interesting model that seems to be ideal for the environment is the one in Co-Reyes et al. [2018] where the model learns from a teacher who provides natural language corrections to the actions taken by the model.

Apart from these, a better meta learning algorithm could be used, like REPTILE Nichol et al. [2018] or a task attentive model agnostic meta learning algorithm like in Jiang et al. [2018]. To further, the claim that the model learns good embeddings for the instructions, the actual embeddings of the instruction could be analyzed.

8 Acknowledgements

We would like to thank Prof. Brenden Lake for his invaluable advice during the framing of the proposal for this project and guiding us towards framing a problem statement. We would also like to thank Ethan Perez for constantly providing us valuable feedback and guidance while implementing the algorithms and importantly introducing us to the environment. All models in the report were trained on GPUs provided by NVIDIA on the CILVR lab cluster, we are grateful for those. Finally, we thank Prof. Ankur Parikh and Anhad Mohananey for guiding us through the course and help us understand the fundamentals of NLP.

References

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Jürgen Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. *Learning a synaptic learning rule*. Université de Montréal, Département d'informatique et de recherche ..., 1990.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

Maxime Chevalier-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. Babyai: First steps towards grounded language learning with a human in the loop. *arXiv preprint arXiv:1810.08272*, 2018.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. *arXiv preprint arXiv:1709.07871*, 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pages 4502–4510, 2016.

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

Ofir Nachum, Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *arXiv preprint arXiv:1805.08296*, 2018.

Théophane Weber, Sébastien Racanière, David P Reichert, Lars Buesing, Arthur Guez, Danilo Jimenez Rezende, Adria Puigdomenech Badia, Oriol Vinyals, Nicolas Heess, Yujia Li, et al. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*, 2017.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Irina Higgins, Arka Pal, Andrei A Rusu, Loic Matthey, Christopher P Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. *arXiv preprint arXiv:1707.08475*, 2017.

Greg Wayne, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja, Agnieszka Grabska-Barwinska, Jack Rae, Piotr Mirowski, Joel Z Leibo, Adam Santoro, et al. Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arXiv:1803.10760*, 2018.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2879–2888, 2018.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pages 3981–3989, 2016.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. URL `http://arxiv.org/abs/1803.02999`.

Luke Metz, Niru Maheswaranathan, Brian Cheung, and Jascha Sohl-Dickstein. Learning unsupervised learning rules. *CoRR*, abs/1804.00222, 2018. URL `http://arxiv.org/abs/1804.00222`.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

Wojciech Zaremba and Ilya Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

Abhishek Gupta, Benjamin Eysenbach, Chelsea Finn, and Sergey Levine. Unsupervised meta-learning for reinforcement learning. *arXiv preprint arXiv:1806.04640*, 2018.

John D Co-Reyes, Abhishek Gupta, Suvansh Sanjeev, Nick Altieri, John DeNero, Pieter Abbeel, and Sergey Levine. Guiding policies with language via meta-learning. *arXiv preprint arXiv:1811.07882*, 2018.

Xiang Jiang, Mohammad Havaei, Gabriel Chartrand, Hassan Chouaib, Thomas Vincent, Andrew Jesson, Nicolas Chapados, and Stan Matwin. On the importance of attention in meta-learning for few-shot text classification. *arXiv preprint arXiv:1806.00852*, 2018.