

---

# e-QRAQ: A Multi-turn Reasoning Dataset and Simulator with Explanations

---

Clemens Rosenbaum<sup>1,2</sup> Tian Gao<sup>1</sup> Tim Klinger<sup>1</sup>

## Abstract

In this paper we present a new dataset and user simulator e-QRAQ (explainable Query, Reason, and Answer Question) which tests an Agent’s ability to read an ambiguous text; ask questions until it can answer a challenge question; and explain the reasoning behind its questions and answer. The User simulator provides the Agent with a short, ambiguous story and a challenge question about the story. The story is ambiguous because some of the entities have been replaced by variables. At each turn the Agent may ask for the value of a variable or try to answer the challenge question. In response the User simulator provides a natural language explanation of why the Agent’s query or answer was useful in narrowing down the set of possible answers, or not. To demonstrate one potential application of the e-QRAQ dataset, we train a new neural architecture based on End-to-End Memory Networks to successfully generate both predictions and partial explanations of its current understanding of the problem. We observe a strong correlation between the quality of the prediction and explanation.

## 1. Introduction

In recent years deep neural network models have been successfully applied in a variety of applications such as machine translation (Cho et al., 2014), object recognition (Krizhevsky et al., 2012; He et al., 2016), game playing (Mnih et al., 2015), dialog (Weston, 2016) and more. However, their lack of interpretability makes them a less attractive choice when stakeholders must be able to understand and validate the inference process. Examples include medical diagnosis, business decision-making and reasoning, le-

gal and safety compliance, etc. This opacity also presents a challenge simply for debugging and improving model performance. For neural systems to move into realms where more transparent, symbolic models are currently employed, we must find mechanisms to ground neural computation in meaningful human concepts, inferences, and explanations. One approach to this problem is to treat the explanation problem itself as a learning problem and train a network to explain the results of a neural computation. This can be done either with a single network learning jointly to explain its own predictions or with separate networks for prediction and explanation. Regardless, the availability of sufficient labelled training data is a key impediment. In previous work (Guo et al., 2017) we developed a synthetic conversational reasoning dataset in which the User presents the Agent with a simple, ambiguous story and a challenge question about that story. Ambiguities arise because some of the entities in the story have been replaced by variables, some of which may need to be known to answer the challenge question. A successful Agent must reason about what the answers might be, given the ambiguity, and, if there is more than one possible answer, ask for the value of a relevant variable to reduce the possible answer set. In this paper we present a new dataset e-QRAQ constructed by augmenting the QRAQ simulator with the ability to provide detailed explanations about whether the Agent’s response was correct and why. Using this dataset we perform some preliminary experiments, training an extended End-to-End Memory Network architecture (Sukhbaatar et al., 2015) to jointly predict a response and a partial explanation of its reasoning. We consider two types of partial explanation in these experiments: *the set of relevant variables*, which the Agent must know to ask a relevant, reasoned question; and *the set of possible answers*, which the Agent must know to answer correctly. We demonstrate a strong correlation between the qualities of the prediction and explanation.

## 2. Related Work

Current interpretable machine learning algorithms for deep learning can be divided into two approaches: one approach aims to explain black box models in a model-agnostic fashion (Ribeiro et al., June 2016; Turner, June 2016); another studies learning models, in particular deep neural networks, by visualizing for example the activations or gradi-

<sup>1</sup>IBM T. J. Watson Research Center, Yorktown Heights, NY, USA <sup>2</sup>University of Massachusetts Amherst, Amherst, MA, USA. Correspondence to: Clemens G. Rosenbaum <cgbr@cs.umass.edu>.

ents inside the networks (Zahavy et al., 2016; Shrikumar et al., 2016; Selvaraju et al., 2016). Other work has studied the interpretability of traditional machine learning algorithms, such as decision trees (Hara & Hayashi, June 2016), graphical models (Kim et al., 2015), and learned rule-based systems (Malioutov & Varshney, 2013). Notably, none of these algorithms produces natural language explanations, although the rule-based system is close to a human-understandable form if the features are interpretable. We believe one of the major impediments to getting NL explanations is the lack of datasets containing supervised explanations.

Datasets have often accelerated the advance of machine learning in their perspective areas (Ferraro et al., 2015), including computer vision (LeCun, 1998; Krizhevsky & Hinton, 2009; Russakovsky et al., 2015; Lin et al., 2014; Krishna et al., 2016), natural language (Lowe et al., 2015; Hermann et al., 2015; Dodge et al., 2015), reasoning (Weston et al., 2015; Bowman et al., 2015; Guo et al., 2017), etc. Recently, natural language explanation was added to complement existing visual datasets via crowd-sourcing labeling (Reed et al., 2016). However, we know of no question answering or reasoning datasets which offer NL explanations. Obviously labeling a large number of examples with explanations is a difficult and tedious task – and not one which is easily delegated to an unskilled worker. To make progress until such a dataset is available or other techniques obviate its need, we follow the approach of existing work such as (Weston et al., 2015; Weston, 2016), and generate synthetic natural language explanations from a simulator.

### 3. The QRAQ Dataset

A QRAQ domain, as introduced in (Guo et al., 2017), has two actors, the User and the Agent. The User provides a short story set in a domain similar to the HomeWorld domain of (Weston et al., 2015; Narasimhan et al., 2015) given as an initial context followed by a sequence of events, in temporal order, and a challenge question. The stories are semantically coherent but may contain hidden, sometimes ambiguous, entity references, which the Agent must potentially resolve to answer the question. To do so, the Agent can query the User for the value of variables which hide the identity of entities in the story. At each point in the interaction, the Agent must determine whether it knows the answer, and if so, provide it; otherwise it must determine a variable to query which will reduce the potential answer set (a “relevant” variable).

In example 1 the actors \$v, \$w, \$x and \$y are treated as variables whose value is unknown to the Agent. In the first event, for example, \$v refers to either Hannah or Emma, but the Agent can’t tell which. In a realistic text this en-

---

#### Example 1 A QRAQ Problem

---

- C1. Hannah and Emma are in the office.
  - C2. John is in the park.
  - C3. Bob and George are in the square.
  - E1. Hannah picks up the gift.
  - E2. \$v goes from the office to the park.
  - E3. \$w goes from the park to the bank.
  - E4. \$x goes from the office to the square.
  - E5. Emma goes from the square to the bank.
  - E6. \$y goes from the square to the bank.
  - Q: Where is the gift?
- 

tity obfuscation might occur due to spelling or transcription errors, unknown descriptive references such as “Emma’s sibling”, or indefinite pronouns such as “somebody”. Several datasets with 100k problems each and of varying difficulty have been released to the research community and are available for download (qra).

## 4. Explainable QRAQ: e-QRAQ

### 4.1. The Dataset

This paper’s main contribution is an extension to the original QRAQ simulator that provides extensive explanations of the reasoning process required to solve a QRAQ problem. These explanations are created dynamically at runtime, in response to the Agent’s actions. The following two examples illustrate these explanations, for several different scenarios: The context (C), events (E), and question (Q) parts of the problem are identical to those in a QRAQ problem. In addition there is a trace of the interaction of a trained Agent (A) model with the User (U) simulator. The simulator provides two kinds of explanations in response to the Agent’s query or answer. The first kind denoted “U” indicates whether the Agent’s response is correct or not and why. The second kind of explanation, denoted “U\*” provides a full description of what can be inferred in the current state of the interaction. In this case the relevant information is the set of possible answers at different points in the interaction (Porch, Boudoir / Porch for Example 2) and the set of relevant variables (\$V0 / none for Example 2).

In Example 2, illustrating a successful interaction, the Agent asks for the value of \$V0 and the User responds with the answer (Silvia) as well as an explanation indicating that it was correct (helpful) and why. Specifically, in this instance it was helpful because it enabled an inference which reduced the possible answer set (and reduced the set of relevant variables). On the other hand, in Example 3, we see an example of a bad query and corresponding critical explanation.

**Example 2** A successfully solved e-QRAQ Problem (with explanations)

*The problem:*

- C1. Silvia is in the porch.
- C2. Charles is in the cellar.
- C3. Maria is in the porch.
- E1. Charles goes from the cellar to the attic.
- E2. Charles goes from the attic to the terrace.
- E3. \$V0 goes from the porch to the boudoir.
- Q: Where is Maria

*The interaction:*

- U\*: Possible Answers: Porch, Boudoir; Relevant Variables: \$V0
- A: Who is \$V0?
- U: \$V0 is Silvia.
- U: This query was helpful, since it allowed the following inference:  
We now know that \$V0 is Silvia, and not Maria. Maria can therefore not be in the boudoir.
- U\*: Possible Answers: Porch; Relevant Variables:  $\emptyset$
- A: Maria is in the porch.
- U: This answer is correct.

**Example 3** An unsuccessfully solved e-QRAQ Problem (with explanations)

*The problem:*

- C1. Paul is in the attic.
- C2. Maria is in the cellar.
- C3. Charles is in the attic.
- E1. Maria goes from the cellar to the terrace.
- E2. \$V4 goes from the attic to the porch.
- E3. Maria goes from the terrace to the boudoir.
- Q: Where is Charles?

*The interaction:*

- U\*: Possible Answers: Attic, Porch; Relevant Variables: \$V4
- A: Who is \$V1?
- U: This query was not helpful, since \$V1 does not even occur in the problem.
- U\*: Possible Answers: Attic, Porch; Relevant Variables: \$V4
- A: Charles is in the porch.
- U: This was a guess, since Charles could still have been \$V4, and thereby in the Porch or in the Attic.  
This guess was correct.

In general, the e-QRAQ simulator offers the following explanations to the Agent:

**Answers** When answering, the User will provide feedback depending on whether or not the Agent has enough information to answer; that is, on whether the set of possible answers contains only one answer. If the Agent has enough information, the User will only provide feedback

on whether or not the answer was correct and on the correct answer if the answer was false. If the agent does not have enough information, and is hence guessing, the User will say so and list all still relevant variables and the resulting possible answers.

**Queries** When querying, the User will provide several kinds of feedback, depending on how useful the query was. A query on a variable not even occurring in the problem will trigger an explanation that says that the variable is not in the problem. A query on an irrelevant variable will result in an explanation showing that the story’s protagonist cannot be the entity hidden by that variable. Finally, a useful (i.e. relevant) query will result in feedback showing the inference that is possible by knowing that variable’s reference. This set of inference can also serve as the detailed explanation to obtain the correct answer above.

The e-QRAQ simulator will be available upon publication of this paper at the same location as QRAQ (qra) for researchers to test their interpretable learning algorithms.

#### 4.2. The “interaction flow”

The normal interaction flow between the User and the Agent during runtime of the simulator is shown in Figure 1, and is - with the exception of the additional explanations - identical to the interaction flow for the original QRAQ

```

User: provides the initial problem state S
while Episode has not terminated do
  Agent: chooses action a and explanation e for S
  if in training mode then
    if in supervised mode then
      User: provides feedback on a and e, and provides the learning targets (i.e. ground truth actions and explanations) a_T, e_T for S
      Agent: trains model on S, <a_T, e_T>
    else if in reinforcement learning mode then
      User: provides feedback on a and e, and provides rewards r_a, r_e for S, a, e
      Agent: trains model on S, <r_a, r_e>
    end if
  end if
  User: computes performance (interaction and explanation accuracies) on a, a_T and e, e_T.
  S ← S', where S' is the state entered from S upon choosing action a
  if action was an answer then
    terminate episode
  end if
end while
    
```

Figure 1. The User-Agent Interaction

problems (Guo et al., 2017). This means that the User acts as a scripted counterpart to the Agent in the simulated e-QRAQ environment. We show interaction flows for both supervised and reinforcement learning modes. Additionally, we want to point out that  $e$  in Figure 1 can be both  $U$  and  $U^*$ , i.e. both the natural language explanation and the internal state explanations. Performance and accuracy are measured by the User, that compares the Agent’s suggested actions and the Agent’s suggested explanations with the ground truth known by the User.

### 5. Experimental Setup

For the experiments, we use the User simulator explanations to train an extended memory network. As shown in Figure 2, our network architecture extends the End-to-End Memory architecture of (Sukhbaatar et al., 2015), adding a two layer Multi-Layer Perceptron to a concatenation of all “hops” of the network. The explanation and response prediction are trained jointly. In these preliminary experiments we do not train directly with the natural language explanation from  $U$ , just the explanation of what can be inferred in the current state  $U^*$ . In future experiments we will work with the  $U$  explanations directly. Specifically, for our

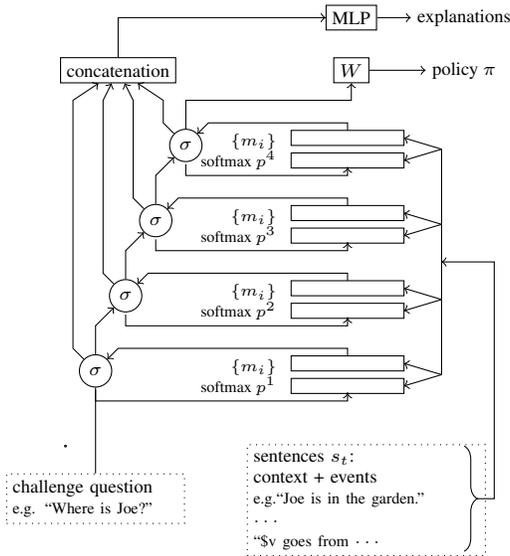


Figure 2. The modified E2E-Memory Network architecture simultaneously generating answers to the challenge question and explanations of its internal belief state, shown with four internal “hops”.

experiments, we provide a classification label for the prediction output generating the Agent’s actions, and a vector  $x_e$  of the following form to the explanation output (where  $oh_d(w)$  is an one-hot encoding of dimensionality (or vo-

cabulary size)  $d$  of word  $w$ , and  $\mathcal{E}$  is the explanation set:

$$x_e = \sum_{w \in \mathcal{E}} oh_d(w) \tag{1}$$

We then train the network, using Adam (Kingma & Ba, 2014), on the combined loss (where  $CE(y, \hat{y})$  is the cross-entropy between the true labels  $y$  and the estimated labels  $\hat{y}$ ,  $\hat{x}_i$  is the network’s interaction output and  $\hat{x}_e$  is the networks explanation output):

$$\mathcal{L} = CE(x_i, \hat{x}_i) + ||x_e - \hat{x}_e||_2^2 \tag{2}$$

For testing, we consider the network to predict a entity in the explanation if the output vector  $\hat{x}_e$  surpasses a threshold for the index corresponding to that entity. We tried several thresholds, some adaptive (such as the average of the output vector’s values), but found that a fixed threshold of .5 works best.

### 6. Results

To evaluate the model’s ability to jointly learn to predict and explain its predictions we performed two experiments. First, we investigate how the prediction accuracy is affected by jointly training the network to produce explanations. Second, we evaluate how well the model learns to generate explanations. To understand the role of the explanation content in the learning process we perform both of these experiments for each of the two types of explanation: relevant variables and possible answers. We do not perform hyperparameter optimization on the E2E Memory Network, since we are more interested in relative performance. While we only show a single experimental run in our Figures, results were nearly identical for over five experimental runs.

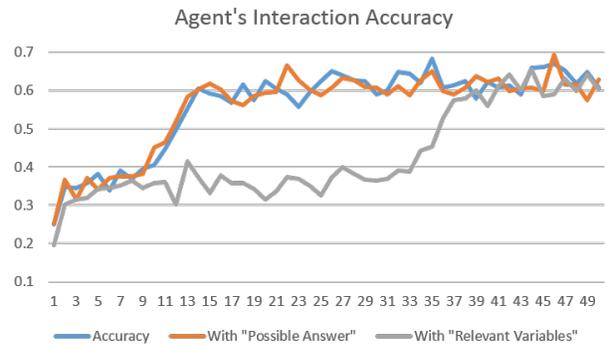


Figure 3. The Interaction Accuracy (over 50 epochs with 1000 problems each)

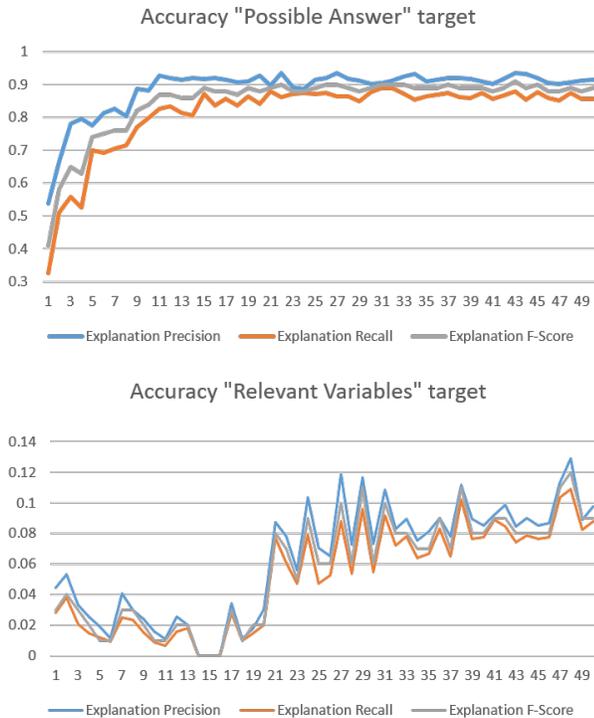


Figure 4. The Explanation Accuracies (over 50 epochs with 1000 problems each)

The experimental results differ widely for the two kinds of explanation considered, where an explanation based on possible answers provides better scores for both experiments. As illustrated in Figure 3, simultaneously learning possible-answer explanations does not affect prediction, while learning relevant-variable explanation learning severely impairs prediction performance, slowing the learning by roughly a factor of four. We can observe the same outcome for the quality of the explanations learned, shown in Figure 4. Here again the performance on possible-answer explanations is significantly higher than for relevant-variable explanations. Possible-answer explanations reach an F-Score of .9, while relevant-variable explanations one of .09 only, with precision and recall only slightly deviating from the F-Score in all experiments.

We would expect that explanation performance should correlate with prediction performance. Since Possible-answer knowledge is primarily needed to decide if the net has enough information to answer the challenge question without guessing and relevant-variable knowledge is needed for the net to know what to query, we analyzed the network’s performance on querying and answering separately. The memory network has particular difficulty learning to query relevant variables, reaching only about .5 accuracy when querying. At the same time, it learns to answer very well,

reaching over .9 accuracy there. Since these two parts of the interaction are what we ask it to explain in the two modes, we find that the quality of the explanations strongly correlates with the quality of the algorithm executed by the network.

## 7. Conclusion and Future Work

We have constructed a new dataset and simulator, e-QRAQ, designed to test a network’s ability to explain its predictions in a set of multi-turn, challenging reasoning problems. In addition to providing supervision on the correct response at each turn, the simulator provides two types of explanation to the Agent: A natural language assessment of the Agent’s prediction which includes language about whether the prediction was correct or not, and a description of what can be inferred in the current state – both about the possible answers and the relevant variables. We used the relevant variable and possible answer explanations to jointly train a modified E2E memory network to both predict and explain it’s predictions. Our experiments show that the quality of the explanations strongly correlates with the quality of the predictions. Moreover, when the network has trouble predicting, as it does with queries, requiring it to generate good explanations slows its learning. For future work, we would like to investigate whether we can train the net to generate natural language explanations and how this might affect prediction performance.

## References

- IBM Research Conversational Reasoning Dataset - “QRAQ”. <http://www.research.ibm.com/cognitive-computing/machine-learning/datasets.html>.
- Bowman, Samuel R, Angeli, Gabor, Potts, Christopher, and Manning, Christopher D. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*, 2015.
- Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- Dodge, Jesse, Gane, Andreea, Zhang, Xiang, Bordes, Antoine, Chopra, Sumit, Miller, Alexander, Szlam, Arthur, and Weston, Jason. Evaluating prerequisite qualities for learning end-to-end dialog systems. *arXiv preprint arXiv:1511.06931*, 2015.
- Ferraro, Francis, Mostafazadeh, Nasrin, Vanderwende, Lucy, Devlin, Jacob, Galley, Michel, Mitchell, Margaret,

- et al. A survey of current datasets for vision and language research. *arXiv preprint arXiv:1506.06833*, 2015.
- Guo, Xiaoxiao, Klinger, Tim, Rosenbaum, Clemens, Bigus, Joseph P, Campbell, Murray, Kawas, Ban, Talamadupula, Kartik, Tesauero, Gerry, and Singh, Satinder. Learning to query, reason, and answer questions on ambiguous texts. *ICLR*, 2017.
- Hara, Satoshi and Hayashi, Kohei. Making tree ensembles interpretable. In *Proc. 2016 ICML Workshop on Human Interpretability in Machine Learning*, pp. 81 – 85, New York, NY, June 2016.
- He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- Hermann, Karl Moritz, Kocisky, Tomas, Grefenstette, Edward, Espeholt, Lasse, Kay, Will, Suleyman, Mustafa, and Blunsom, Phil. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pp. 1693–1701, 2015.
- Kim, Been, Finale, Doshi-Velez, and Shah, Julie. Mind the gap: A generative approach to interpretable feature selection and extraction. In *Neural Information Processing Systems*, 2015.
- Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. URL <https://arxiv.org/abs/1412.6980>.
- Krishna, Ranjay, Zhu, Yuke, Groth, Oliver, Johnson, Justin, Hata, Kenji, Kravitz, Joshua, Chen, Stephanie, Kalantidis, Yannis, Li, Li-Jia, Shamma, David A, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.
- Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.
- Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- LeCun, Yann. The mnist database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>, 1998.
- Lin, Tsung-Yi, Maire, Michael, Belongie, Serge, Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C Lawrence. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755. Springer, 2014.
- Lowe, Ryan, Pow, Nissan, Serban, Iulian, and Pineau, Joelle. The ubuntu dialogue corpus: A large dataset for research in unstructured multi-turn dialogue systems. *arXiv preprint arXiv:1506.08909*, 2015.
- Malioutov, Dmitry and Varshney, Kush. Exact rule learning via boolean compressed sensing. In *International Conference on Machine Learning*, pp. 765–773, 2013.
- Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Rusu, Andrei A, Veness, Joel, Bellemare, Marc G, Graves, Alex, Riedmiller, Martin, Fidjeland, Andreas K, Ostrovski, Georg, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- Narasimhan, Karthik, Kulkarni, Tejas, and Barzilay, Regina. Language understanding for text-based games using deep reinforcement learning. *arXiv preprint arXiv:1506.08941*, 2015.
- Reed, Scott, Akata, Zeynep, Lee, Honglak, and Schiele, Bernt. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 49–58, 2016.
- Ribeiro, Marco Tulio, Singh, Sameer, and Guestrin, Carlos. Model-agnostic interpretability of machine learning. In *Proc. 2016 ICML Workshop on Human Interpretability in Machine Learning*, pp. 91 – 95, New York, NY, June 2016.
- Russakovsky, Olga, Deng, Jia, Su, Hao, Krause, Jonathan, Satheesh, Sanjeev, Ma, Sean, Huang, Zhiheng, Karpathy, Andrej, Khosla, Aditya, Bernstein, Michael, Berg, Alexander C., and Fei-Fei, Li. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.
- Selvaraju, Ramprasaath R, Das, Abhishek, Vedantam, Ramakrishna, Cogswell, Michael, Parikh, Devi, and Batra, Dhruv. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*, 2016.
- Shrikumar, Avanti, Greenside, Peyton, Shcherbina, Anna, and Kundaje, Anshul. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*, 2016.
- Sukhbaatar, Sainbayar, Weston, Jason, Fergus, Rob, and others. End-to-end memory networks. In *Advances in neural information processing systems*, pp. 2440–2448, 2015. URL <http://papers.nips.cc/paper/5846-end-to-end-memory-networks>.

Turner, Ryan. A model explanation system: Latest updates and extensions. In *Proc. 2016 ICML Workshop on Human Interpretability in Machine Learning*, pp. 1 – 5, New York, NY, June 2016.

Weston, Jason. Dialog-based language learning. *arXiv preprint arxiv:1604.06045*, 2016.

Weston, Jason, Bordes, Antoine, Chopra, Sumit, Rush, Alexander M, van Merriënboer, Bart, Joulin, Armand, and Mikolov, Tomas. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015.

Zahavy, Tom, Ben-Zrihem, Nir, and Mannor, Shie. Gray-ing the black box: Understanding dqns. *arXiv preprint arXiv:1602.02658*, 2016.