Introduction:

The goal of this project was to implement a deep learning model that was capable of learning any NLP problem by utilizing episodic memory. By restructuring any given NLP task into a question-answer problem, the model should ideally be able to generate the text sequence answer, where the format of the answer is dependent on the context of the type of question asked.

Data Description:

The data used for this project was Facebook's bAbI task set. This dataset consist of twenty different tasks. Each task contains 3 parts: the supporting information, the question, and the answer.

The supporting information is a series of text broken into a new line per each sentence. The number of input context vectors can be variable. Some instances have only a single sentence while others have multiple. Not all supporting context sentences are required or relevant in determining the final answer; some of the sentences are unnecessary noise. While the bAbI dataset also contains additional arguments which indicate which lines of the supporting context are relevant, this was not used in our training process.

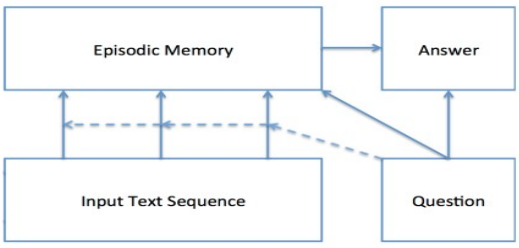
The question context is a single text question. Generally, the format of the question for any given task remains the same, with the exception of either the subject, direct, or indirect object of the sentence changing between training instances. For example, in the single supporting sentence training set, the format of the questions are: Where is [X], where X can change between instances – Where is [John, Mary, the football, etc.]? Each task has a different format of question.

The training data contained the expected answer to the asked question. All of the bAbI task only expect a single word output, however, our model should be capable of generating a sequence of output words.

The format of each task's context, question, and answer format are included in Appendix A with a single training instance to show the general format.

Methods Used:

The model used follows the architecture described in the paper *Ask Me Anything: Dynamic Memory Network for Natural Language Processing (2015)* [Kumar, Irsoy, et. al]. The model itself consist of four module components: the input context, the question context, the memory unit, and the answer unit.



<u>0. Preprocessing</u>

The training data used in the model is all text based. Before the data could be used with the model, the input text was tokenized and mapped to a numerical value. Each text sequence was broke up by each word. Additionally, the end of sentence punctuation (either the period for input context or question mark for the question context) was separated out into its own token. Four special tokens were also added; <s> and </s> were used to separate between sentences in the input context when there were multiple input context vectors, the <UNK> token was used in any instance where a word was not part of the vocabulary, and the <PAD> token was used to make make all input vectors the same length. The maximum input length is just the maximum number of words per sentence plus any additional added tokens over the full training input file. A vocabulary was created from all unique

tokens in the input training file and each token was mapped to a numerical value by just assigning the tokens an integer value based on the token's index in the vocabulary in sequential order.

1. Input Module

The input module consists of two major components; the word embedding layer and the context encoding RNN.

The input to this module is the set of vectorized input sentence contexts. For each value in the input vector, the embedding layer creates a one-to-many mapping from vector value vocab index to a fixed length vector of EMBEDDING_SIZE, where EMBEDDING_SIZE is a model hyper-parameter. The intent of this layer is to the learn the relationship between words rather than just the words themselves based on index value. Kumar, Irsoy, et. al used the gloVe word embedding in their original model rather than training their own embedding layer, however, in our implementation we do not use transfer learning for the word embeddings and include the embedding layer during training.

The context encoding RNN uses a gated recurrent neural network (GRU). We apply the GRU at each time step (word embedding) such that our intermediary output has the shape MAX_INPUT_LENGTH by RNN_UNITS, were each time step is the hidden weight values of the GRU at the given iteration. For the intermediary input module outputs, we only care about the hidden states at the end of each sentence rather than the hidden states after each word or the hidden states of the padding. During preprocessing, we tracked the index of end of sentence tokens for each set of input context. After applying the input encoder GRU, we parse out only the hidden states at a time step where there is a end of sentence token. Since each input context is variable length and can have a different number of context sentences, we apply zero mask padding such that the final output has a length of MAX_CONTEXT_SENTENCES, were the first N time steps are the end of sentence hidden layer outputs and the remainder are zero vectors.

2. Question Module

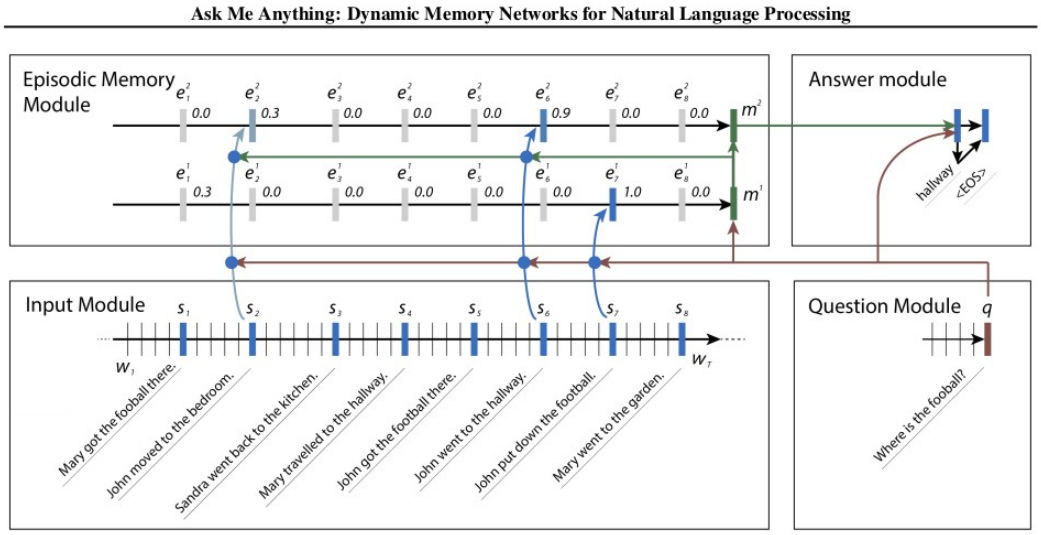
Like the Input Module, the Question Module has two components; the word embedding layer and a context encoding RNN. The word embedding layer is shared between both the input context and question context modules. Since the question context is always a single sentence, when we apply the question encoder GRU to question context word embeddings, we only keep the final GRU output and ignore the hidden layer values.

3. Memory Unit

The memory unit consist of two components; an attention mechanism and a memory encoder GRU. These two components are applied multiple times, where the number of loops is the number of episodes. The idea for the memory unit is to mimic how people think and reevaluate information as it receives additional context. Each episode, the attention mechanism focuses on a different part of the input context with respect to the original question. In theory, the number of required episodes should be the number of sentences containing the information needed to answer the question. Since the amount of episodic memory is finite, the attention mechanism is trained to only preserve the context states that are relevant in answering the initial question and ignore irrelevant information. The attention mechanism works by applying a weight that is proportional to the probability of an input being relevant given the question context and previous memory values. Irrelevant information passes low input values to the memory encoder GRU, which cause little to change between memory output time steps. Each episode the memory encoder GRU passes over the entire input context sequence, therefore, the total number of time steps applied to the memory unit overall is the number of episodes times the number of encoded input time steps. The attention mechanism is only applied at the start of each episode, so the total number of calls to the attention layer is the same as the number of episode loops. The final memory output for the final time step of the final episode is used as an input to the answer module.

4. Answer Unit

The final answer component consists of a single decoder GRU. The number of time steps the GRU layer is applied is equal to the number of output sequences we want to generate. Given that for our training and testing data, all the answers are a single word, we only apply one time step of the answer decoder GRU. The inputs to the answer decoder GRU is a concatenation of the previous output with the encoded question context, given the GRU's previous hidden state, which is initialized to the memory modules final output. The answer decoder GRU uses the softmax activation function between time step outputs. The final results is a word embedding of the probability of each word in the vocab, where the argmax out the output corresponds to the index of the word in the vocab that should be outputted.

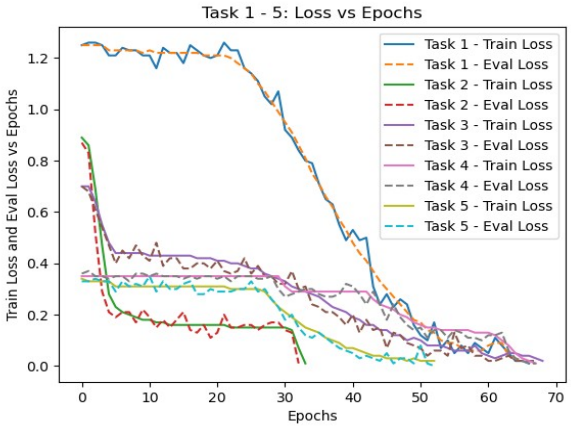


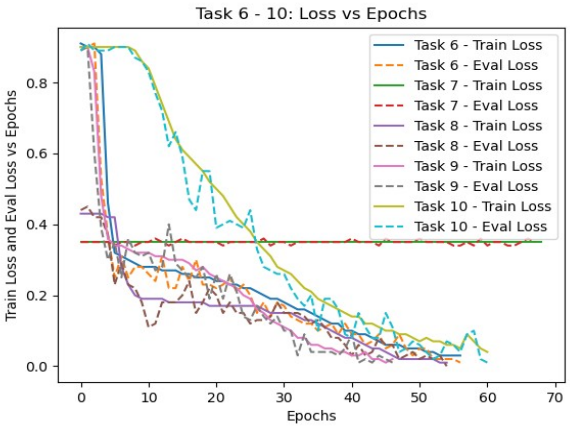
Hyper-parameters and Training:

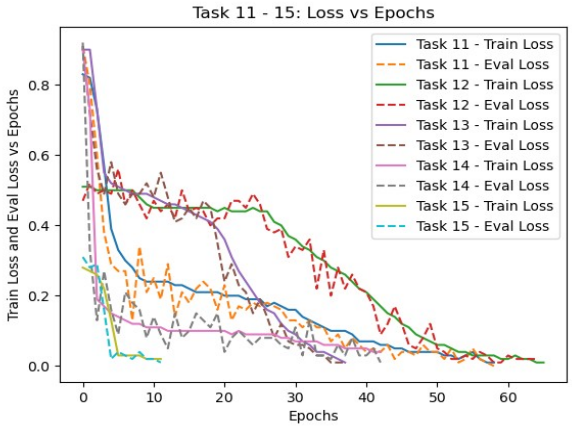
Given the amount of computational resources required to do a single pass of the entire model, we were unable to meaningfully tune the input parameters. However, we use the recommended hyper-parameters and model settings used by Kumar, Irso, et. al. Each of the GRU's use the same number of hidden layer RNN units, that is, EMBEDDING_SIZE and RNN_UNITS are the same value across all modules. Additionally, early stopping is used as a form of regularization. During training, we use a dropout rate of 0.1 between the word embedding and n GRU encoder layers for the input and question context modules.

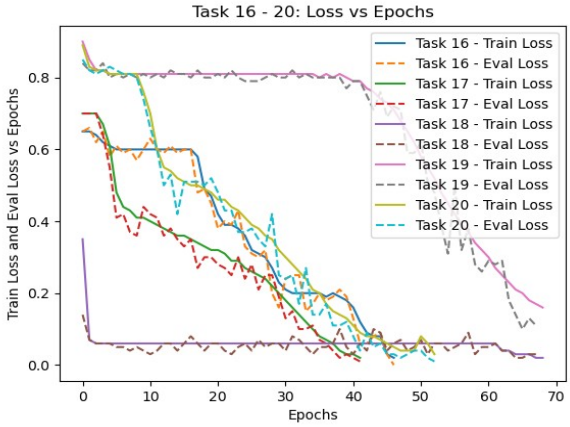
Hyper-parameter	Value
HIDDEN RNN UNITS	80
LEARNING RATE	0.001
EPOCHS	70
EPISODES	3
BATCH SIZE	64
EARLY STOP THRESHOLD	0.01
OPTIMIZER	ADAM
ADAM beta0	0.9
ADAM beta1	0.999

Loss vs Epochs for Task 1-20

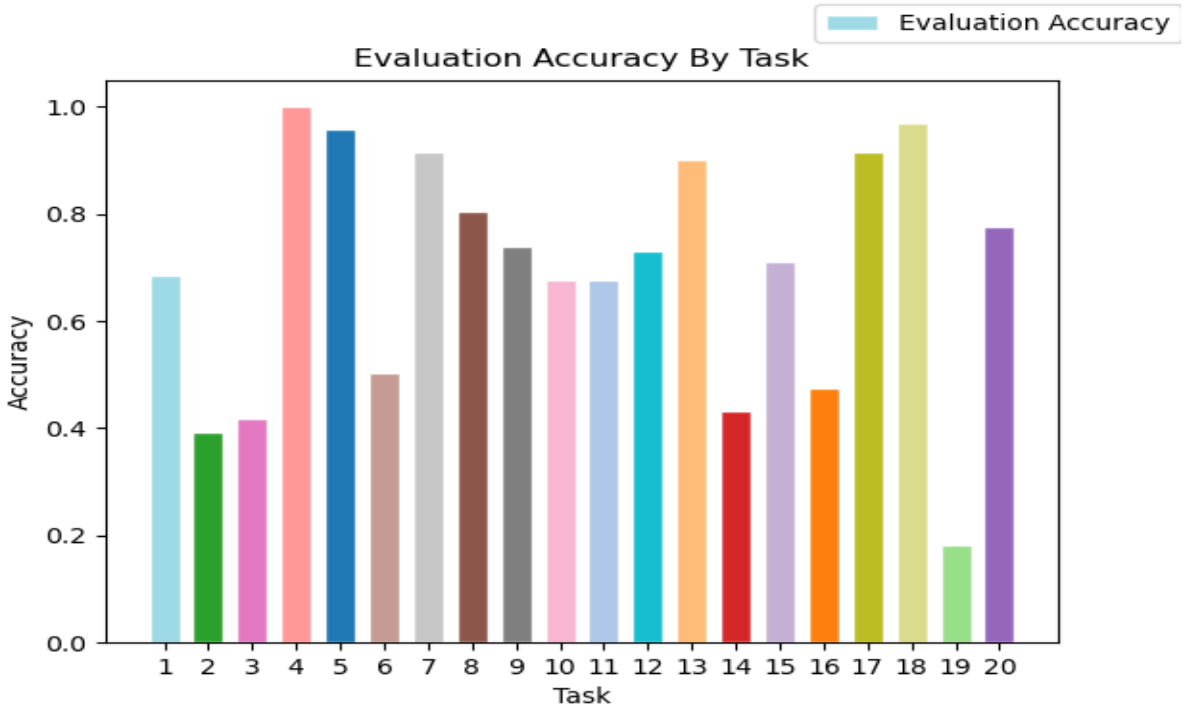


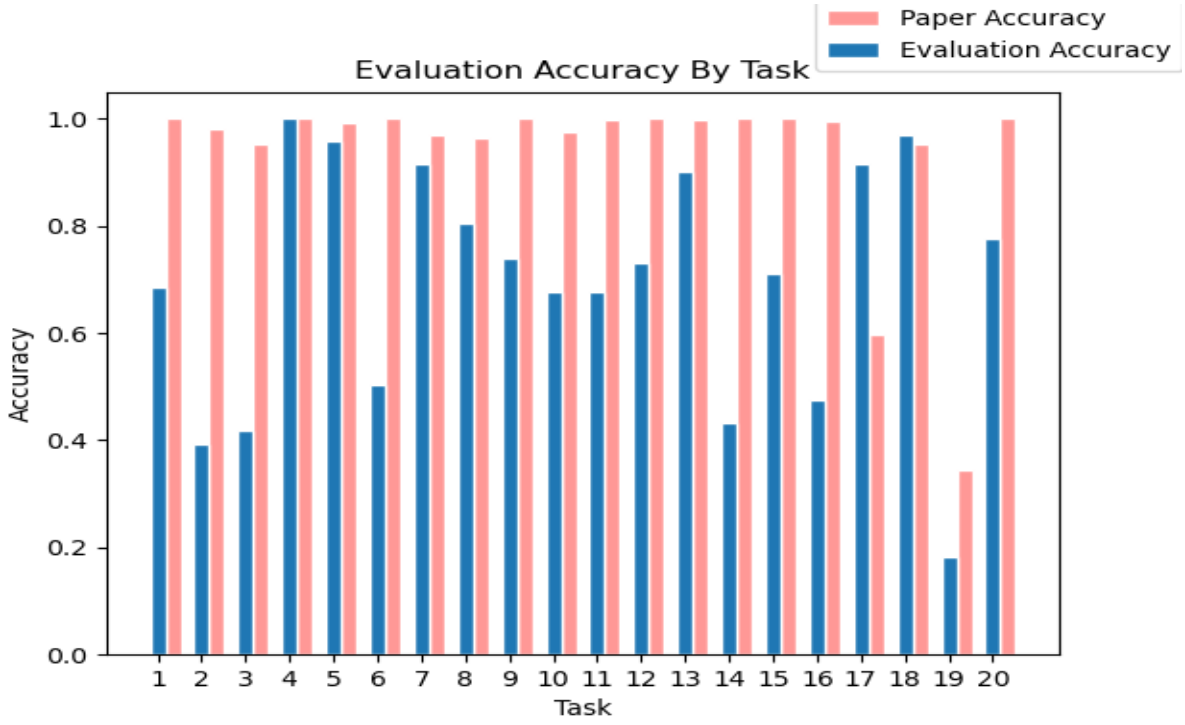






Accuracy Per Task





Results:

After training each task for 70 epochs, we achieve the following results:
Note: Early stopping with a tolerance of 0.01 was used, so not all task were trained for the total 70 epochs

Task	Name	Epochs	Train-loss	Evaluation Loss	Evaluation Accuracy
1	WhereIsActor	68	0.01	0.02	68.6
2	WhereIsObject	35	0.01	0.01	39.3
3	WhereWasObject	70	0.02	0.01	41.9
4	IsDir	69	0.01	0.01	100.0
5	WhoWhatGave	54	0.02	0.0	95.9
6	IsActorThere	58	0.03	0.01	50.3
7	Counting	70	0.35	0.35	91.4
8	Listing	56	0.01	0.0	80.4
9	Negation	48	0.02	0.01	74.0
10	Indefinite	62	0.04	0.01	67.7
11	BasicCoreferecne	60	0.01	0.0	67.6
12	Conjunction	67	0.01	0.02	73.2
13	CompoundCoreferecnce	39	0.01	0.01	90.0
14	Time	44	0.04	0.01	43.1
15	Deducation	13	0.02	0.01	71.2
16	Induction	48	0.02	0.0	47.6
17	PositionalReasoning	43	0.02	0.01	91.5
18	Size	70	0.02	0.03	97.0
19	PathFinding	70	0.16	0.11	18.2
20	Motivations	54	0.03	0.01	77.7

Analysis of Results:

Overall, our model was able to preform well on some task, however, there were a few task were the model was not able to learn anything at all. When comparing the number of epochs ran to the training and evaluation loss vs evaluation accuracy, the model would not have necessarily preformed any better had it be ran for longer. For example, task 2, 6, 14, and 16 all reached a training loss that was low enough to trigger early stopping, however, their evaluation accuracy was still low and the model was not able to generalize well for these task. In order for our model to better perform on these specific task, we would have had to increase the number of hidden GRU units for the input, question, and memory modules. Additionally training the memory unit for more episodes may also help with these task. Despite the model potentially being insufficiently large to learn certain task, it was still able to perform well on other task, even given its smaller size.

By comparison, task 19 still only had an evaluation accuracy of 18.2% after the allowed 70 epochs; looking at the individual loss plot for task 19, the evaluation and training loss remained relatively constant for the first 40 epochs, after which the training error started to decrease. Had task 19 been trained for more epochs, we may have seen the overall accuracy improve. For other task, such as 1, 8, 9, 10, 11, 12, 15, and 20 which performed moderately well but did not achieve a 90%+ evaluation accuracy, our early stopping regularization may have been too strict. Had we used a lower tolerance for early stopping or a different method of regularization entirely, we may potentially had seen an increase from mid 70-80% accuracy to 90%+. However, most likely the model itself was too small and was only able to learn partial trends, without having a sufficient episodic memory size to completely learn the trend for the specific tasks.

Task 7 clearly hit a saddle point during training; the training error remained constant and the evaluation error oscillated with some variance around the training error. Despite this, the model still performed well on this task.

Given a different initialization or using a training method more capable of dealing with saddle points, the model could potential achieve better results for this specific task in its current state.

Overall, the model performed well despite its reduced size. Given that the majority of task that did not perform well still converged prior to reaching 70 epochs leads us to believe that our network size was too small, rather than an insufficient amount of training. Other than increasing the network size, a potential area of improvement would be to switch to using a pre-trained word embedding layer. The original paper used gloVe for their word embedding layer, however we trained our own word embeddings with an embedding layer size of 80, whereas the gloVe word embedding layer is size 500. Switching to gloVe would provide the benefit of a larger embedding layer size in addition to reducing the number of network parameters required to be trained. Having fewer parameters to train would reduce the overall training time required and allow more progress to be made per epoch on the remaining trainable parameters.

The individual loss plots are included in the project documents.

APPENDIX A: EXAMPLE DATA

==========

1. Basic factoid QA with single supporting fact (WhereIsActor)

Example Question:
- Where is John?

Example Context:
- John travelled to the hallway.
- Mary journeyed to the bathroom.

Example Answer:
hallway

==========

2. Factoid QA with Two Supporting Facts (WhereIsObject)

Example Question:
- Where is the milk?

Example Context:
- Mary got the milk there.
- John moved to the bedroom.
- Sandra went back to the kitchen.
- Mary travelled to the hallway.

Example Answer:
hallway

==========

3. Factoid QA with Three Supporting Facts (WhereWasObject)

Example Question:
- Where was the apple before the bathroom?

Example Context:
- John grabbed the apple.
- John went to the office.
- Sandra took the milk.
- John journeyed to the bathroom.
- John went to the garden.
- John discarded the apple there.

Example Answer:
office

==========

4. Two Argument Relations: Subject vs Object (IsDir)

Example Question:
- What is the bathroom east of?

Example Context:
- The hallway is east of the bathroom.
- The bedroom is west of the bathroom.

Example Answer:
bedroom

==========

5. Three Argument Relations (WhoWhatGave)

Example Question:
- What did Fred give to Jeff?

Example Context:
- Fred picked up the football there.
- Fred gave the football to Jeff.

Example Answer:
football

==========

6. Yes/No Questions (IsActorThere)
Example Question:
- Is John in the kitchen?

Example Context:
- Mary got the milk there.
- John moved to the bedroom.

Example Answer:
no

==========

7. Counting (Counting)

Example Question:
- How many objects is Mary carrying?

Example Context:
- Mary got the milk there.
- John moved to the bedroom.

Example Answer:
one

==========

8. List/Sets (Listing)

Example Question:
- What is Mary carrying?

Example Context:
- Mary got the milk there.

- John moved to the bedroom.

Example Answer:
milk

==========
9. Simple Negation (Negation)

Example Question:
- Is Sandra in the bedroom?

Example Context:
- John is in the hallway.
- Sandra is in the kitchen.

Example Answer:
no

==========
10. Indefinite Knowledge (Indefinite)

Example Question:
- Is Mary in the school?

Example Context:
- Mary moved to the kitchen.
- Mary is either in the school or the kitchen.
- Bill is either in the park or the bedroom.
- Bill is in the school.

Example Answer:
maybe

==========
11. Basic Coreference (BasicCoreference)

Example Question:
- Where is John?

Example Context:
- John journeyed to the hallway.
- After that he journeyed to the garden.

Example Answer:
garden

==========
12. Conjunction (Conjunction)

Example Question:
- Where is Mary?

Example Context:
- John and Mary travelled to the hallway.
- Sandra and Mary journeyed to the bedroom.

Example Answer:
bedroom

==========
13. Compound Coreference (CompoundCoreference)

Example Question:
- Where is John?

Example Context:
- John and Mary went back to the hallway.
- Then they went to the bathroom.

Example Answer:
bathroom

==========
14. Time Manipulation (Time)

Example Question:
- Where was Bill before the office?

Example Context:
- This morning Mary moved to the kitchen.
- This afternoon Mary moved to the cinema.
- Yesterday Bill went to the bedroom.
- Yesterday Mary journeyed to the school.
- Where was Mary before the cinema? kitchen 2 1
- Yesterday Fred went back to the cinema.
- Bill journeyed to the office this morning.

Example Answer:
bedroom

==========
15. Basic Deduction (Deduction)

Example Question:
- What is emily afraid of?

Example Context:
- Wolves are afraid of mice.
- Sheep are afraid of mice.
- Winona is a sheep.
- Mice are afraid of cats.
- Cats are afraid of wolves.
- Jessica is a mouse.
- Emily is a cat.
- Gertrude is a wolf.

Example Answer:
wolf

==========
16. Basic Induction (Induction)

Example Question:
- What color is Brian?

Example Context:
- Lily is a swan.
- Bernhard is a lion.
- Greg is a swan.
- Bernhard is white.
- Brian is a lion.
- Lily is gray.
- Julius is a rhino.
- Julius is gray.
- Greg is gray.

Example Answer:
white

==========
17. Positional Reasoning (PositionalReasoning)

Example Question:
- Is the pink rectangle to the right of the red square?

Example Context:
- The pink rectangle is to the left of the triangle.
- The triangle is to the left of the red square.

Example Answer:
no

==========
18. Reasoning About Size (Size)

Example Question:
- Does the box fit in the chocolate?

Example Context:
- The suitcase fits inside the box.
- The chocolate fits inside the box.
- The container is bigger than the box of chocolates.
- The container is bigger than the suitcase.
- The box is bigger than the box of chocolates.
- The container is bigger than the chocolate.
- The chocolate fits inside the container.
- The chocolate fits inside the suitcase.

- The chocolate fits inside the chest.
- The suitcase fits inside the container.

Example Answer:
no

==========

19. Path Finding (PathFinding)

Example Question:
- How do you go from the bathroom to the hallway?

Example Context:
- The garden is west of the bathroom.
- The bedroom is north of the hallway.
- The office is south of the hallway.
- The bathroom is north of the bedroom.
- The kitchen is east of the bedroom.

Example Answer:
s,s

==========

20. Reasoning About Agent’s Motivation (Motivations)

Example Question:
- Why did jason go to the kitchen?

Example Context:
- Jason is thirsty.
- Where will jason go? kitchen
- Jason went to the kitchen.

Example Answer:
thirsty