– Independent Work Report Fall, 2016 –

Exploring the Efficacy of Dynamic Memory Networks for Question-Answering Tasks in Harry Potter Fiction

Avinash Nayak
Adviser: Christiane D. Fellbaum

Abstract

The need and desire for information among people, and specifically answers to questions about general knowledge or specific academia, permeates throughout the world. Current information search and QA methods require large amounts of human resources, and hence automated QA is a significant landmark in the fields of Artificial Intelligence and Natural Language Processing. This paper explores and analyzes the efficacy of Dynamic Memory Networks, which are a memory-based neural network model, in being able to answer questions about both simple and complicated language input from a variety of sources. According to the work done in [1], and based off of the YerevaNN github [2], a dynamic memory network is implemented on the bAbI dataset, and it has extended and modified to be implemented on not only Microsoft's MCTest dataset, but also a Harry Potter dataset. New modules are added to the existing implementation of the DMN from YerevaNN github [2] repository, and new methods of analyzing more intricate text are described. This paper serves as a benchmark for future work in automated QA for more intricate pieces of literature in the areas of philosophy, history, religion, and fan-fiction.

1. Introduction and Problem Background

Today, human beings crave information at their fingertips, but they lack the mental resources to store large amounts of information in their own brains for practical use, since humans have low memory capacity as compared to collections of computers. As a result, the study of Question-Answer (QA) tasks is increasingly relevant; QA remains to be one of the most important components of the broader study of natural language processing (NLP). Automated solutions to QA tasks can enhance findings in various other research fields such as those of Turing Tests and Human-Computer

Interactions, and furthermore, can advance human-computer information seeking through tools like Microsoft's Cortona or Apple's Siri.

This work takes on the challenge of building an end-to-end automated QA task solver using machine learning, instances of which have been constructed before, but it also analyzes a new realm of QA tasks. Rather than solely addressing simple-sentence QA tasks, such as those found in instruction guidebooks, or from general knowledge on the web, the ultimate goal of this project is to analyze the accuracy of dynamic memory networks (DMNs, to be discussed later) on more complex literature-based data sets, including textual samples from Harry Potter novels. The entrance of QA into the realm of literature with more complex sentence structures can have largely positive consequences on information retrieval from not only internet-based general knowledge sources, but also historical literary works, and any piece of text or writing that has been written with single sentences expressing many ideas each. Automated QA for advanced literature can highly facilitate the ability of extraction of concrete and crucial information about the large set of literary texts from the past, fictional and non-fictional. The novels of the Harry Potter series serve as a starting point from which more research can be continued later on QA for advanced literature.

2. Related Work and Challenges of Current Work

While the DMN model is currently the most popular model in addressing QA tasks, there has been work in the past few years relating to automated machine-learning based models that attempt to solve the QA problem for appropriately structured datasets. Within Weston et al. [3], the idea of memory networks is presented as a method by which to utilize a long-term memory component as a QA knowledge base, with text as output. With respect to recent work, Kumar et al. [1] explores the performance of the DMN model on Facebook's bAbI dataset, which consists of 20 unique QA tasks, which are detailed later in this paper. Raghuvanshi et al. [4] also makes an attempt at implementing a DMN in tensorflow for bAbI tasks.

3. Approach

Most papers in the areas of QA so far only address QA tasks based on simple sentences and scenarios, such as "Mary went to eat lunch. She entered the cafeteria." As such, this analysis of fiction with more advanced syntactic structure (i.e. prepositional phrases, appositive phrases, compound sentences, participial phrases, and more) is the primary novelty in this paper; from this point on, the simple sentence structure dataset that is used for initial training and testing is the Facebook bAbI dataset, for which a supervised DMN is built. For each of the different formats of data included within the dataset (i.e. Single Supporting Fact, Lists/Sets, Simple Negation, etc.), testing accuracy is measured over many epochs of training. Contrastingly, the complex sentence structure dataset used will be Microsoft's MCTest (Machine Comprehension Test) dataset, consisting of a collection of compound-structure fictional stories, along with answer choices to questions about the stories. Then, different textual samples, each resembling the MCTest data format mentioned above, have been collected from each of the Harry Potter novels to test QA accuracy for questions based on the novels' passages.

In order to incorporate the complex sentence structure structure of the input data from MCTest as well as Harry Potter, two primary modifications/additions had to be made to the DMN used in Kumar et al. [1]. First, an choice selection module was added in order to incorporate the multiple-choice format that was desired for answering questions about Harry Potter text samples. Secondly, the memory module and answer module were modified such that they corresponded with the added choice selection module that was mentioned earlier. The ideas of these modifications for MCTest and Harry Potter data are derived from Qian et al. [5], and they are described in greater detail in the implementation section of this report. Module addition and modification served as the technical novelty of this project, and dataset restructuring (bAbI tasks were given textual alterations to have them more resemble complex literature, and MCTest stories were converted to Harry Potter passages) served as an ideological novelty, both of which can be explored more in future research.

Although an approach has been outlined above, it remains to be discussed why the novelty of introducing more complex sentence data is crucial to building automated QA task solvers within NLP. If we currently look at the Facebook bAbI dataset, it relies on a limited vocabulary, and as has been alluded to before, contains a very basic sentence structure. The intricacy of not only the English language, but of the writing of literature in general, is the reason why more advanced datasets are crucial to work with - they serve as the start point for automated QA being able to eventually address more rigorous forms of literature.

It is to be noted that there are philosophical works by authors like Henry David Thoreau, Karl Marx, or Aristotle, that have advanced sentence structures as well. However, when children are reading as they grow up, the first and most fundamental instances of intricate sentence structure occur in young adult fiction books, and for that reason, data from a fiction book was used in this project - in order to serve as a starting point for future research in QA task solving with intricate non-fiction literature.

4. Data

Every segment of a QA task involves a portion of textual information that is to be parsed, known as the knowledge base (KB), whose form could be of varying length, ranging from one sentence to more than one paragraph. Then, a question is asked, and the DMN lets the AI solver produce either a single word answer or a selection from multiple choices of answers. The following two lines represent the two datasets used and modified, with certain portions of MCTest being replaced by text found in Harry Potter novels (not bAbI because it has a very minimal defined vocabulary):

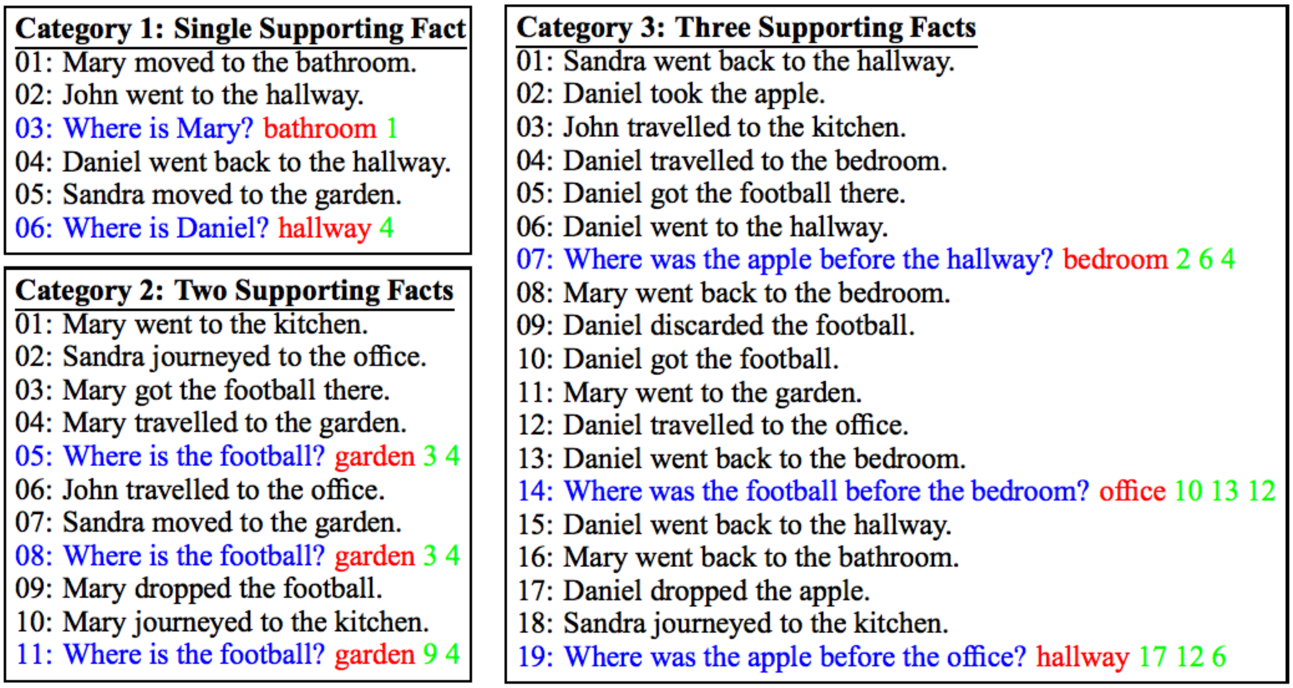
Facebook bAbI QA: https://research.fb.com/projects/babi

Microsoft MCTest QA: http://research.microsoft.com/en-us/um/redmond/projects/mctest

4.1. Facebook bAbI QA

Structurally, the bAbI dataset contains 20 types of tasks, and each task is associated with 1000 training questions and 1000 test questions. Each file has 3000 lines total of text, and with each line containing either a sentence of input or a question in bAbI, it is evident that there are hence

2000 lines of sentence inputs, and hence the ratio of input sentences to questions is 2:1. However, congruence is not necessary, so questions may come in bursts as the file is parsed, and may be consecutive as well. Figure 1 is a sample from a text file that depicts how input-question-answer tuples may be ordered. The format is unmodifiable - all input files for bAbI tasks must have a collection of sentences on separate lines, followed by a question, an associated answer, and line references for facts leading up to the answer:



Category 1: Single Supporting Fact
01: Mary moved to the bathroom.
02: John went to the hallway.
03: Where is Mary? bathroom 1
04: Daniel went back to the hallway.
05: Sandra moved to the garden.
06: Where is Daniel? hallway 4

Category 2: Two Supporting Facts
01: Mary went to the kitchen.
02: Sandra journeyed to the office.
03: Mary got the football there.
04: Mary travelled to the garden.
05: Where is the football? garden 3 4
06: John travelled to the office.
07: Sandra moved to the garden.
08: Where is the football? garden 3 4
09: Mary dropped the football.
10: Mary journeyed to the kitchen.
11: Where is the football? garden 9 4

Category 3: Three Supporting Facts
01: Sandra went back to the hallway.
02: Daniel took the apple.
03: John travelled to the kitchen.
04: Daniel travelled to the bedroom.
05: Daniel got the football there.
06: Daniel went to the hallway.
07: Where was the apple before the hallway? bedroom 2 6 4
08: Mary went back to the bedroom.
09: Daniel discarded the football.
10: Daniel got the football.
11: Mary went to the garden.
12: Daniel travelled to the office.
13: Daniel went back to the bedroom.
14: Where was the football before the bedroom? office 10 13 12
15: Daniel went back to the hallway.
16: Mary went back to the bathroom.
17: Daniel dropped the apple.
18: Sandra journeyed to the kitchen.
19: Where was the apple before the office? hallway 17 12 6

Figure 1: Derived from [6]. A collection of task samples, where a question's answer has one, two, or three supporting facts from prior inputs. With regards to format, note that sample inputs are in black, questions are in blue, answers are in red, and information supporting references are in green.

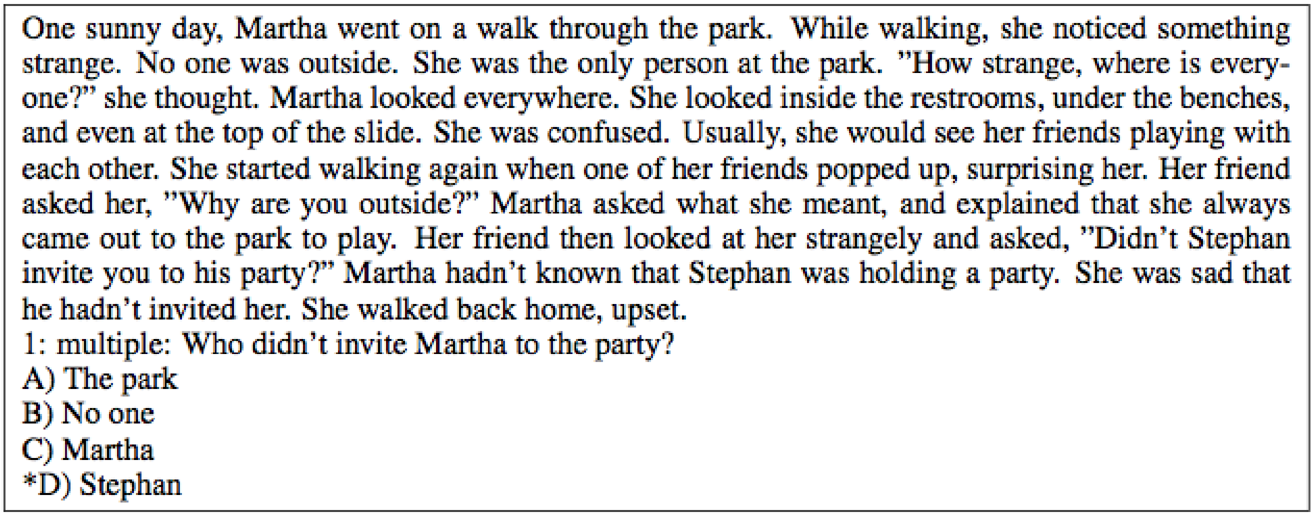
As is noticeable from the sample text above, the input sentences of the bAbI dataset follow a very straightforward sentence format - the majority of them will simply have a subject, verb, and a direct object or predicate nominative (sometimes a prepositional phrase). Sentences are neither compound, complex, nor compound-complex in structure, and they do not contain more intricate structures like prepositional phrases, gerunds, participial phrases, appositives, or other words and phrases that defy the standard subject-verb format of sentences. In contrast, the MCTest dataset contains more complicated sentences and paragraphs, and while the bAbI dataset is presented such that objective questions (who, what, when, where) are to be asked, MCTest is presented for subjective questions as well (why, how).

As explained above, the bAbI dataset serves as a baseline for a supervised DMN implementation, but in this paper, the novelty in using bAbI was that it was altered in certain ways to allow it to follow a less predictable format. As is presented in the Results and Evaluation section of this paper, changes in training and testing accuracy are analyzed, with respect to comparing the results from the original bAbI dataset to the ones from the altered datasets.

4.2. Microsoft MCTest QA

The MCTest dataset files consists of text blocks, where each block contains a made-up story of a paragraph's length (between 150 and 300 words), four multiple choice questions for each story (pertaining on multiple sentences from it), and four multiple choices for each question, along with each correct answer choice. The formatting of the dataset is derived from the the following paper: Qian et al. [5].The corpus itself has two different sets of stories: MC160 (160 stories) and MC500 (500 stories). For train, development, and test sets, MC160 is randomly divided into 70, 30, and 60 stories respectively, and the MC500 is randomly divided into 300, 50, and 150 stories respectively. The development set is used to minimize over-fitting; hyper-parameter tuning was performed on the development set (10 % of the training set randomly selected). Figure 2 is an example of a story, along with an example of one multiple choice question following it.

As can be noted from the MCTest sample data, the sentences and groups of sentences in the story contain syntactic elements that are more complicated than those of the bAbI inputs. Prepositional phrases like "at the park" and "inside the restrooms", and participial phrases such as "surprising her" are examples of more complex syntactic elements existing in the input, but because of them, the input more closely resembles a real portion of a fiction novel, or an online encyclopedia, from which automated QA would be much desired in the future. One caveat is that multiple choices are given, so the QA task turns into a problem of identification and selection, rather than providing a textual answer. However, this is first step towards automated QA for more advanced language input.



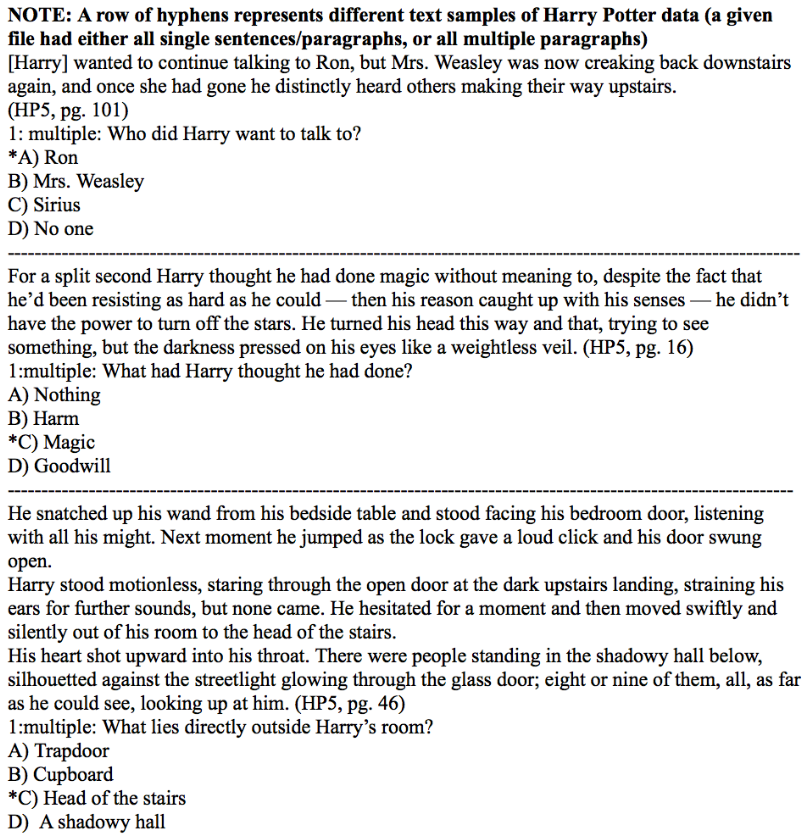
One sunny day, Martha went on a walk through the park. While walking, she noticed something strange. No one was outside. She was the only person at the park. ”How strange, where is everyone?” she thought. Martha looked everywhere. She looked inside the restrooms, under the benches, and even at the top of the slide. She was confused. Usually, she would see her friends playing with each other. She started walking again when one of her friends popped up, surprising her. Her friend asked her, ”Why are you outside?” Martha asked what she meant, and explained that she always came out to the park to play. Her friend then looked at her strangely and asked, ”Didn’t Stephan invite you to his party?” Martha hadn’t known that Stephan was holding a party. She was sad that he hadn’t invited her. She walked back home, upset.
1: multiple: Who didn’t invite Martha to the party?
A) The park
B) No one
C) Martha
*D) Stephan

Figure 2: A MCTest sample story with one multiple choice question.

4.3. Harry Potter dataset

The constructed Harry Potter dataset files had similar structure to MC160 and MC500 in format: the files consisted of text blocks containing a passage from any of the seven Harry Potter novels, four multiple choice questions for each passage, and four options for each multiple choice question, with the correct answer marked. However, while all MCTest passages were between 150 to 300 words, the goal of using the Harry Potter passages was to test different levels of machine comprehension with varying levels of text-spacial granularity. Hence, the original MC160 stories were substituted with 160 passages of varying lengths of one sentence/paragraph (1̃5-75 words), and two to three paragraphs (1̃00-200 words) - MC500 stories were cumbersome to alternate with Harry Potter passages, since the lengths of the paragraphs needed to be very specific, and there was not enough data to have a varied distribution among the novels. An example of these substitutions with varying spatial granularities (passage lengths) is presented in figure 3.

It is evident that the sentences from Harry Potter book 5 in the example above are not as clear and crisp (they are contextual) as the sentences in MCTest stories, but they follow a more advanced sentence structure than those in bAbI, and hence they are quite relevant. Questions test a variety of topics from factual identification to positional reasoning of objects or locations, to multi-sentence logical connection, resembling the variety of questions that MCTest itself has. Additionally, for the



NOTE: A row of hyphens represents different text samples of Harry Potter data (a given file had either all single sentences/paragraphs, or all multiple paragraphs)

[Harry] wanted to continue talking to Ron, but Mrs. Weasley was now creaking back downstairs again, and once she had gone he distinctly heard others making their way upstairs.
(HP5, pg. 101)
1: multiple: Who did Harry want to talk to?
*A) Ron
B) Mrs. Weasley
C) Sirius
D) No one

For a split second Harry thought he had done magic without meaning to, despite the fact that he'd been resisting as hard as he could — then his reason caught up with his senses — he didn't have the power to turn off the stars. He turned his head this way and that, trying to see something, but the darkness pressed on his eyes like a weightless veil. (HP5, pg. 16)
1:multiple: What had Harry thought he had done?
A) Nothing
B) Harm
*C) Magic
D) Goodwill

He snatched up his wand from his bedside table and stood facing his bedroom door, listening with all his might. Next moment he jumped as the lock gave a loud click and his door swung open.
Harry stood motionless, staring through the open door at the dark upstairs landing, straining his ears for further sounds, but none came. He hesitated for a moment and then moved swiftly and silently out of his room to the head of the stairs.
His heart shot upward into his throat. There were people standing in the shadowy hall below, silhouetted against the streetlight glowing through the glass door; eight or nine of them, all, as far as he could see, looking up at him. (HP5, pg. 46)
1:multiple: What lies directly outside Harry's room?
A) Trapdoor
B) Cupboard
*C) Head of the stairs
D) A shadowy hall

Figure 3: Three sample Harry Potter passages with one example multiple choice question each (the first two are of a single sentence/paragraph length and the last one is of a multi-paragraph length)

sake of consistency and shared use between the supervised DMN implementation built for MCTest and Harry Potter both in this paper, if a word in a passage or sample from Harry Potter was not part of the MCTest data corpus, it was substituted with a closely semantically similar word that did appear in the MCTest set of 8000 words, and the substitution needed to have the same part of speech (questions and answers were asked accordingly of the adjusted passages). All validation of accuracy of answers to these questions was supported by common sense judgments from the text, as well as supervised research and fact-checking on Pottermore - a digital information base of the Harry Potter world. Future work could incorporate a more automated validation approach, but that will be discussed in the summary of this report.

5. Implementation

The initial supervised DMN implementation (used solely for bAbI and bAbI modification) represents the one used in Kumar et al. [1], and is based on YerevanNN's DMN implementation with Theano (a Python library allowing for efficient mathematical computation with multi-diminensional arrays) and Lasagne (a library to build and train neural networks in Theano). This DMN model consists of four distinct modules, each serving its own purpose, explained in the first four subsections of this Implementation section. In the latter subsections, the novelty and modifications of this paper are explained.

5.1. Input Module

The function of this module is to be able to encode raw textual inputs from the given task into a set of distributed vector representations, and this process is completed via a recurrent neural network, or RNN as introduced by Elman et al. [7]. From the given vocabulary, within bAbI, MCTest, and Harry Potter text segments, word embeddings are generated by Stanford's GloVe, an unsupervised learning algorithm for obtaining vector representations of words [8] - the embeddings form a matrix, since each word has an associated numerical vector, and every associated vocabulary has a list of words. These generated embeddings are fed into the RNN, and at each time step t, the RNN updates its hidden state representation, $h_t = RNN(L[w_t], h_{t-1})$, in which L is the word embedding matrix, and w_t represents the respective word index of the tth word of the input sentence. The choice of RNN used is the gated recurrent unit network (GRU) [9], and the internal mechanics of the GRU are described in equations 1 to 4.

5.2. Question Module

Similar to the structure of the input module, the question module also follows an encoding process of converting the task's questions into distributed vector representation. This representation is repeatedly processed by a RNN, and the hidden state representation at a certain time t is $q_t = RNN(L[{w_t}^Q], q_{t-1})$, where L is the same word embedding matrix as used for the input module, and

w_t^Q represents the respective word index of the tth word of the input question. The final hidden state representation is the input for the episodic memory module (to be described), and is the basis by which the episodic memory module iterates.

The internal mathematical updates of the GRU for the input, question, episodic memory, and answer modules lies below, and is derived from Kumar et al. [1], where each time step t has input x_t and hidden state h_t from the input module:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1} + b^{(z)}) \tag{1}$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1} + b^{(r)}) \tag{2}$$

$$h_t' = \tanh(Wx_t + r_t \circ Uh_{t-1} + b^{(h)}) \tag{3}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ h_t' \tag{4}$$

In equations 1 to 4, $\circ$ is the Hadamard product, or element-wise product of two vectors, and W and U are parameter matrices while b is a parameter vector. $W^{(z)}$, $W^{(r)}$, $W \in \mathbb{R}^{n_H \times n_I}$ and $U^{(z)}$, $U^{(r)}$, $U \in \mathbb{R}^{n_H \times n_H}$; n_H and n_I are hyper-parameters. From this point on in the paper, the GRU computation as demonstrated in equations 1 to 4 can be abbreviated in the following format $h_t = GRU(x_t, h_{t-1})$.

5.3. Episodic Memory Module

As alluded to before, this module iterates over representations from the input and question module, and it selects which portions of input to focus on through an attention mechanism, which is to be described. Then, it produces a memory vector representation, which is a function of the given question(s) and the previous memory. The memory vector m^i shall be updated as follows: $m^i = GRU(e^i, m^{i-1})$, and the initial state of the GRU is the question vector itself, so $m^0 = q$, where q is the final vector state representation of the RNN encoder within the question module.

Attention Mechanism:

The preliminary attention mechanism allows the episodic memory module to choose which parts of the input to focus on. The implementation follows the one described in Kumar et al. [1]; based on a candidate fact c_t that serves as a potential one-word answer for the bAbI question, a previous memory vector m^{i-1}, and a question vector q for the gate computation, we define the gate as pass i to be $g_t^i = G(c_t, m^{i-1}, q)$, where G is a scoring function from [1] - it is a two-layer feed forward neural network. As for the stopping criteria, the neural network update stops at input T_M when the attention mechanism has scored an end-of-pass candidate fact c^* as the highest relevance answer over all the input sentences (this was the stopping criteria only for the bAbI tasks, not for MCTest and Harry Potter fiction).

5.4. Answer Module

The preliminary answer module is derived from Kumar et al. [1], and uses the memory vector from the halting input, m^{T_M} as an initial hidden state representation, along with question vector q and previous prediction to generate an answer via GRU. All four modules are summarized by Figure 4.

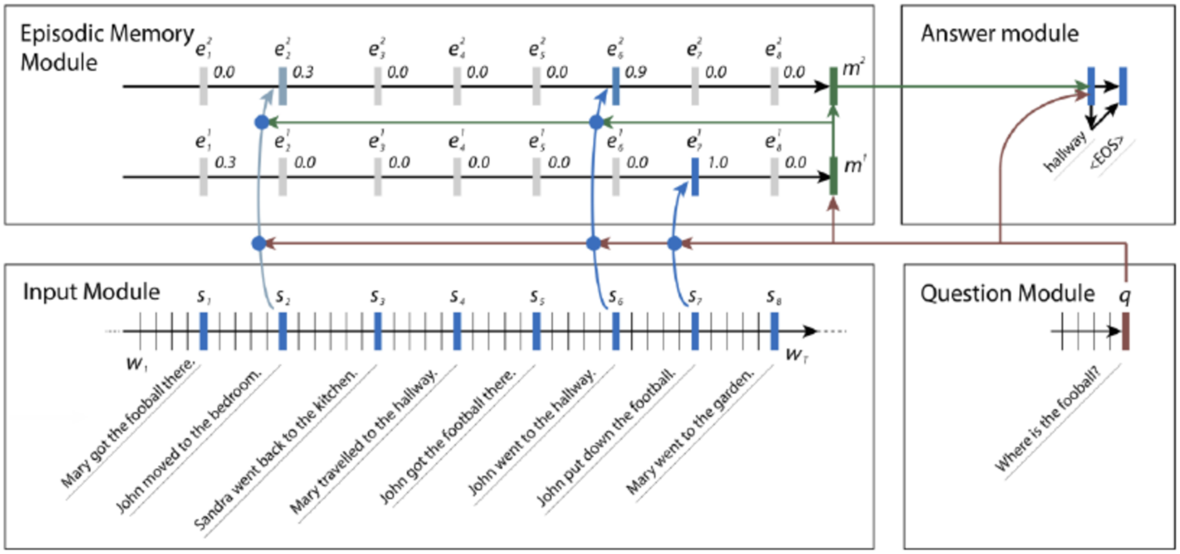


Figure 4: A visualization of the DMN from [1]. The diagram depicts a real input sentence list and the attention gates triggered in the episodic memory module. The value e_n^m represents the episode corresponding with input sentence S_n at pass m.

5.5. Added GRU Instance - Choice Selection Module

The MCTest Dataset and the newly created Harry Potter dataset have a contrasting format from bAbI. For bAbI QA tasks, every query is correlated with an answer of only a single word, but for MCTest and the Harry Potter dataset, the question results in four option choices that selection is performed from, and in addition, not every choice consists of only a single word answer. Hence, this selection module was constructed to read the corresponding question-answer information as input. A GRU (acting as part of a RNN), following the same mathematical modeling as described in [1], was utilized in order to retrieve a distributed hidden state vector representation of all four choices together. The final representation of all four choice vectors, (Ca, Cb, Cc, Cd) was outputted in a matrix format.

5.6. Memory/Answer Module Alteration

While the DMN's episodic memory module for bAbI QA functioned based on its dependence on one-word answers for training and testing, it needed to be modified for incorporating the choice-selection format of the Harry Potter fiction passage dataset, as well as MCTest. Within the attention mechanism, in Kumar et al. [1], the feature vector z captures a function input, question, and prior memory, but the different in output in this project is that there are feature vectors for each of the four different choices for each question, so these need to be appropriately piped in to the answer module.

Another point of contrast between bAbI and the Harry Potter/MCTest dataset format is that while the answer for bAbI can only be one word, the answer for Harry Potter and MCTest can be up to a sentence. The reason is that for bAbI, given its simplicity of primarily containing simple sentences with a subject, predicate, and direct object, solely "who", "what", and "when" questions are addressed, but for more complicated pieces of literature starting with Harry Potter (and MCTest), "where", "how", and "why" questions are asked, requiring more than a single word answer of a noun or an adjective. Here, we introduce a novel matrix A, and if we let C be our choice matrix with the four column vectors being choice vectors of the four answer choices to any question, then through matrix multiplication, we can compute the value of $m_{mem}^{T}AC$, where m_{mem} is the output from the episodic memory module. This computation yields a 4-dimensional row-vector, where the maximum element, after appropriate training on A, will yield the correlation between each option (a through d) and the prior memory - the highest value index will be the answer choice to select once iterations have halted.

The process described above is a measurement of cosine similarity of each of the four output choices with memory vector that is outputted by the episodic memory module.

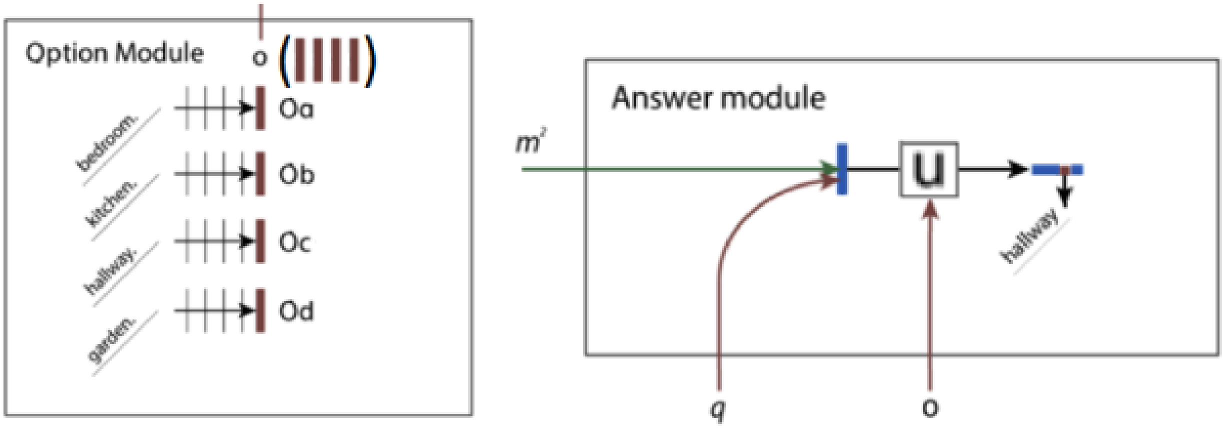


Figure 5: An example of the correlation of an option module with an answer module from [5]. The diagram depicts an option state representation o, and the u within the small box in the answer module represents a cosine similarity detection of the most similar option to past memory.

6. Results and Evaluation

Running the DMN for bAbI, MCTest, and Harry Potter datasets was indeed a computationally expensive task that also required sufficient disk space for creation and manipulation of the RNN structures that were used for the input, question, and episodic memory modules. Although the DMN was not able to run on my personal computer, fortunately, Princeton's computing resources were valuable in their assistance with this predicament. Known concisely as Princeton's "Cycle Servers", the CS Department has four runnable Intel based machines for department users, and each machine is a Fujitsu RX200 S8 server with dual eight-core 2.8GHz Intel Xeon E5 2680 v2 processors with 256GB of RAM, running Linux. Running the bAbI DMN on this server, via SSH, took about 1 minute for initial training setup and about 30 seconds per epoch of training and accuracy testing.

Running the DMN for MCTest took about 8 minutes for 20 epochs, and running the DMN for Harry Potter took about 7 minutes for 20 epochs of training for the multi-paragraph data and 4 minutes for the single sentence/paragraph data.

6.1. Intricacy Segmentation of bAbI Tasks

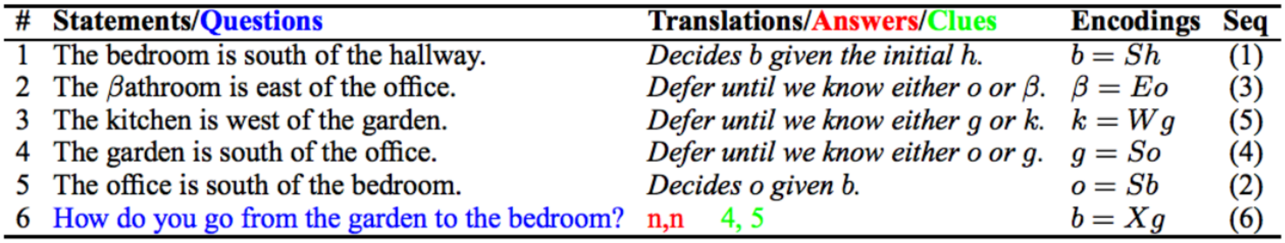
In table 1 are the twenty bAbI tasks, and their converging accuracies with the supervised DMN from Kumar et al. [1] (the accuracies are from extensive training done by the Kumar model - the model for this paper has not been trained for enough epochs to have as high accuracies as [1], but the accuracies from [1] serve as a benchmark for segmentation of tasks based on their ability to reach a testing accuracy of higher than 95 %).

bAbI task	**Testing Accuracy %**
Single Supporting Fact	100
Two Supporting Facts	98.2
Three Supporting Facts	95.2
Two Argument Relations	100
Three Argument Relations	99.3
Yes/No question	100
Counting	96.9
Lists/Sets	96.5
Simple Negation	100
Indefinite Knowledge	97.5
Basic Co-reference	99.9
Conjunction	100
Compound Co-reference	99.8
Time Reasoning	100
Basic Deduction	100
Basic Induction	99.4
Positional Reasoning	59.6
Size Reasoning	95.3
Path Finding	34.5
Agent’s motivations	100
Average value	**93.6**

Table 1: The preliminary testing accuracies for bAbI tasks as represented in Kumar et al. [1].

As depicted in table 1, testing accuracies differ for tasks, and intriguingly there is a linguistic reason for this differentiation, the idea of which is derived from [6]. Specifically with regards to the tasks with less than 95 % testing accuracy, the intricacy of Positional Reasoning and Path Finding is more than that of Single Supporting Fact or Simple Negation, for example, because of the contrast between uni-relations and multi-relations. The characteristic of being uni-relational can be epitomized as the inability of an object existing in more than one location or circumstance at the same time, and single supporting fact and simple negation are prime examples of this because, for example in task 1 within table 1, each question has only one answer that is directly identifiable by the episodic memory module, which is the best and most optimal one. However, what makes positional reasoning and path finding challenging to achieve high accuracy on is that these tasks are cumulative, and multiple facts from different portions of the input sentences can correlate together in some order to form a final answer. While the episodic memory module solely parses in one direction (from the last or most recent sentence or event, up until the first), multi-directional traversal is lacking. For example, in Path Finding within Figure 6, as the statements are traversed in sequence starting from "The bedroom is south of the hallway" to "The office is south of the bedroom", for 100 % human accuracy, it is necessary to defer certain pieces of information about relative locations of the bathroom, kitchen, and garden until all necessary information is revealed.

At this point in the paper, we shall define two difficulty based tiers that each of the tasks may lie into. Tier 1 tasks are those out of the 20 that have higher than 95 % accuracy from table 1, and tier 2 tasks are those tasks that do not lie in tier 1. The idea of splitting into tiers or into "passing" vs. "failing" tasks is derived from Kumar et al. [1].



#	Statements/Questions	Translations/Answers/Clues	Encodings	Seq
1	The bedroom is south of the hallway.	*Decides b given the initial h.*	$b = Sh$	(1)
2	The βathroom is east of the office.	*Defer until we know either o or β.*	$\beta = Eo$	(3)
3	The kitchen is west of the garden.	*Defer until we know either g or k.*	$k = Wg$	(5)
4	The garden is south of the office.	*Defer until we know either o or g.*	$g = So$	(4)
5	The office is south of the bedroom.	*Decides o given b.*	$o = Sb$	(2)
6	How do you go from the garden to the bedroom?	n,n 4, 5	$b = Xg$	(6)

Figure 6: Input Sample for Path Finding from [6]

6.2. bAbI Experimentation

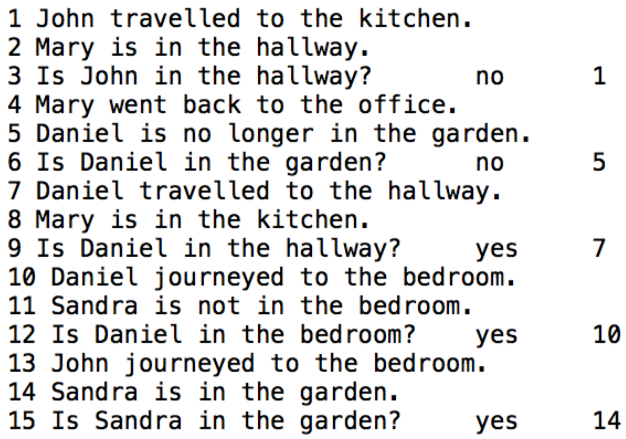
Based on the varied difficulty of achieving high levels of accuracy on the 20 QA tasks, certain changes were made to specific portions of bAbI that would lie within the bAbI vocabulary, but would make simpler sentences more complex. This idea correlates with the goal of this paper as a whole, which is to analyze the efficacy of QA on more intricate inputs. The goal of the bAbI experimentation, beyond the preliminary results given by [1], was to determine which bAbI tasks are affected by these alterations, and whether there lies a significant difference in how much the accuracy of tasks of different tier levels.

An important note is that because only a maximum of 20 epochs were used, full convergence was not yet reached, so some final accuracy values for QA tasks might be slightly lower than they are in table 1.

6.2.1. Tense Shifting : The first dataset alteration for bAbI took place in the form of tense shifts. This specific form of modification was selected because changes in tense from verbs like "have" to "had", or "is" to "was", still lie within bAbI's vocabulary. These tense shifts are applied in different levels to the bAbI task inputs (normal vs. 13 % sentences altered in simple negation - 2/15 in each block - and 20 % sentences altered in positional reasoning - 2/10 in each block), and the accuracy of the controlled vs. altered distributions is measured and plotted (the difference in accuracy over alteration is normalized by either 13 or 20 to make the changes comparable in simple negation vs. positional reasoning). Altered sentences were selected by random in each block, with the requirement that they must yield consistence for the given block (for example, if the last sentence in figure 8 was changed to past tense, line 14 was changed to past tense as well since it correlates with the answer of "yes" to the query on line 15).

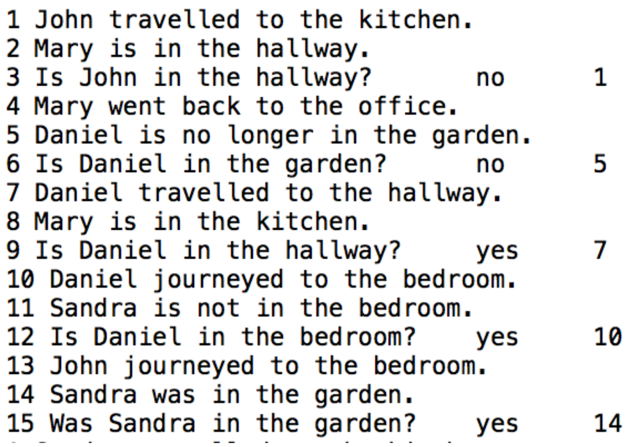
The motivation behind tense shifting as a data alteration technique was that writing, especially fictional writing, can have significant tense alterations, specifically in dialogue. It is extremely important, within the work of automated QA for complex writing, to detect tenses appropriately in order to answer accurately about questions. As an example, figures 7 and 8 represent original (on the left) and altered (on the right) samples of a simple negation block in training data - note that solely the last sentence is changed as a test of how bAbI will perform in each circumstance, but the choice for which sentence(s) in each block that is altered is random. The original and altered samples of a positional reasoning block are corresponding, as it is that with them as well, two correlated sentences in each block of sentence length 10 have a switch in tense.

For both simple negation and positional reasoning, training and testing accuracy were measured over 20 epochs, and were plotted for both the original bAbI dataset and the modified one with tense alteration (13 %). Figures 9 and 10 depict accuracy plots for Simple Negation and Positional Reasoning, respectively, over 20 epochs of training and testing accuracy, and their different values over different levels of input alteration.



```
1 John travelled to the kitchen.
2 Mary is in the hallway.
3 Is John in the hallway?       no      1
4 Mary went back to the office.
5 Daniel is no longer in the garden.
6 Is Daniel in the garden?      no      5
7 Daniel travelled to the hallway.
8 Mary is in the kitchen.
9 Is Daniel in the hallway?     yes     7
10 Daniel journeyed to the bedroom.
11 Sandra is not in the bedroom.
12 Is Daniel in the bedroom?    yes     10
13 John journeyed to the bedroom.
14 Sandra is in the garden.
15 Is Sandra in the garden?     yes     14
```

Figure 7: Original Simple Negation Task sample block



```
1 John travelled to the kitchen.
2 Mary is in the hallway.
3 Is John in the hallway?       no      1
4 Mary went back to the office.
5 Daniel is no longer in the garden.
6 Is Daniel in the garden?      no      5
7 Daniel travelled to the hallway.
8 Mary is in the kitchen.
9 Is Daniel in the hallway?     yes     7
10 Daniel journeyed to the bedroom.
11 Sandra is not in the bedroom.
12 Is Daniel in the bedroom?    yes     10
13 John journeyed to the bedroom.
14 Sandra was in the garden.
15 Was Sandra in the garden?    yes     14
```

Figure 8: Altered Simple Negation Task sample block

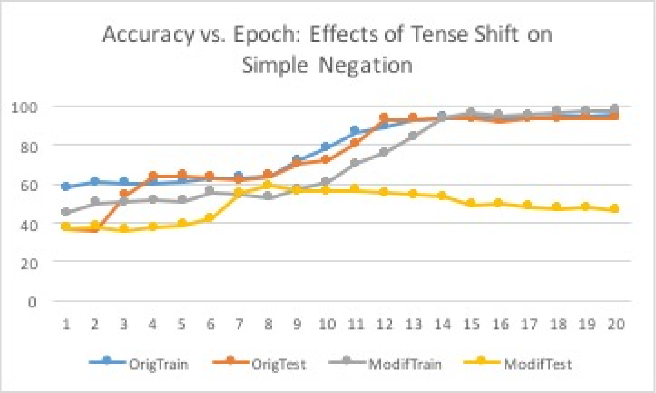


Figure 9: Changes in training and testing accuracy over 20 epochs in Simple Negation task (Tense Shift)

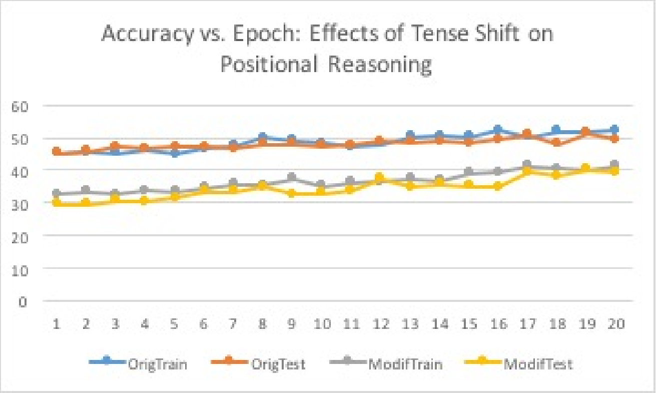
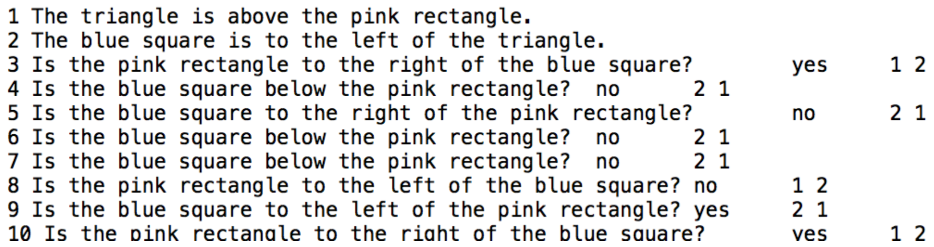


Figure 10: Changes in training and testing accuracy over 20 epochs in Positional Reasoning (Tense Shift)

In figures 9 and 10, the blue and orange dotted lines are the original training and testing accuracies, and the gray and yellow dotted lines are the modified training and testing accuracies, respectively. An analytical comparison of the two plots (tier 1 difficulty task on the left and tier 2 on the right) depicts that while the modified Positional Reasoning accuracy lies generally below the original for both testing and training data, the Simple Negation does not follow this trend - instead, modified training has an extremely high accuracy exceeding even the original training after 20 epochs, but the testing accuracy has a significant dropoff in value after around 10 epochs. The reason for this difference in outcome between tier 1 task modification and tier 2 task modification is that the Simple Negation task (and most tier 1 tasks) allow overfitting within tense shifts. Much of the tier 1 training data has present and past tense already included within ("Mary went back to the office" and "Mary is in the hallway" exist in the original data itself), and hence the RNN model that is being trained is likely to conform heavily to dependency on the specific tense shifts being performed. For this reason, it is probable that the RNN trained specifically to recognize present and past tense in the already existing training data, and hence the randomness of tense shifts applied to the testing data would have been detected less easily. In contrast, positional reasoning and path finding are more logical than factual or situational, so they have less to do with tense, and more with traversal and logical connection of many facts to provide a conclusion. This is likely the reason why the have a standard drop in both training and testing after data alteration, without one being more or differently affected than the other after tense shifting. Due to the stark contrast between the effects of tense shifting on tier 1 vs. tier 2 tasks (Simple Negation vs. Positional Reasoning), normalization of 13 % and 20 % alteration is not accounted for in this subsection, but it is in the next one.

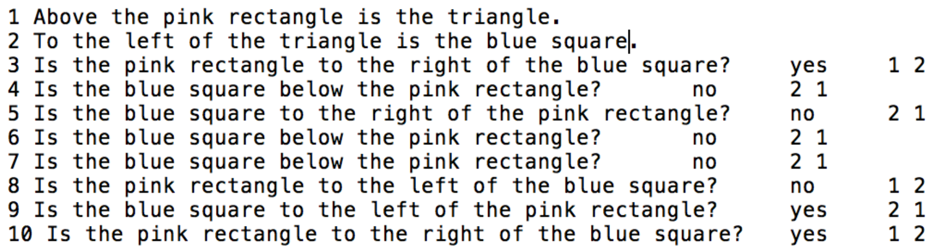
6.2.2. Word ordering : Similar to tense switches, altered word ordering is a common appearance in fiction and more complex literature. Often, for the sake of dramatization and change from monotonous tone, instead of a sentence such as "the boy sat under the tree", an author could write, "Under the tree, sat the boy" or "Beneath the tree, the boy sat". For that sake, the effect of word ordering on training and testing accuracy of the bAbI DMN is also explored in this paper.

Both samples coming from the positional reasoning task training data, Figure 11 is a sampled



```
1 The triangle is above the pink rectangle.
2 The blue square is to the left of the triangle.
3 Is the pink rectangle to the right of the blue square?      yes     1 2
4 Is the blue square below the pink rectangle?  no      2 1
5 Is the blue square to the right of the pink rectangle?      no      2 1
6 Is the blue square below the pink rectangle?  no      2 1
7 Is the blue square below the pink rectangle?  no      2 1
8 Is the pink rectangle to the left of the blue square? no      1 2
9 Is the blue square to the left of the pink rectangle? yes     2 1
10 Is the pink rectangle to the right of the blue square?     yes     1 2
```

Figure 11: Original Positional Reasoning Order



```
1 Above the pink rectangle is the triangle.
2 To the left of the triangle is the blue square.
3 Is the pink rectangle to the right of the blue square?    yes     1 2
4 Is the blue square below the pink rectangle?       no     2 1
5 Is the blue square to the right of the pink rectangle?    no      2 1
6 Is the blue square below the pink rectangle?       no     2 1
7 Is the blue square below the pink rectangle?       no     2 1
8 Is the pink rectangle to the left of the blue square?     no      1 2
9 Is the blue square to the left of the pink rectangle?     yes     2 1
10 Is the pink rectangle to the right of the blue square?   yes     1 2
```

Figure 12: Modified Positional Reasoning Order

original version of a block of QA and Figure 12 is a modified version of the QA block. It is to be noted that the order of words has swapped, such that there is no deviation outside the bounds of the bAbI vocabulary, but the motivation as mentioned before is that the sentence sounds less monotonous and plain than it did in the original version. Sentences of more unpredictable structure are the ones that differentiate adult fiction and more intricate writing from children's books, and hence serve as one of the transition points to applying automated QA to literature of higher complexity. Figures 13 and 14 serve as the equivalents of figures 9 and 10, except for the fact that they depict accuracy change due to word-reordering, rather than tense shift . After examination of figure 14, it is evident

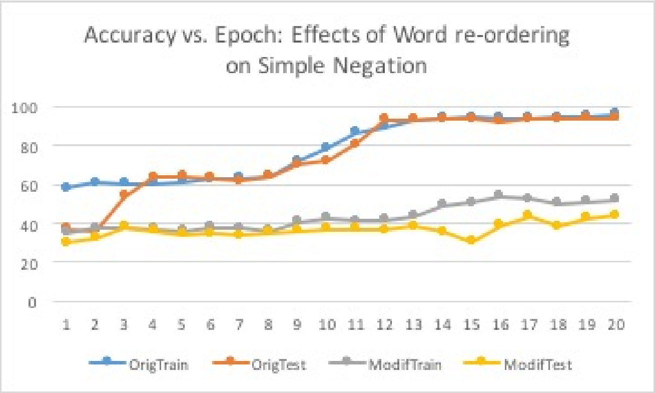


Figure 13: Changes in training and testing accuracy over 20 epochs in Simple Negation task (Word re-ordering)

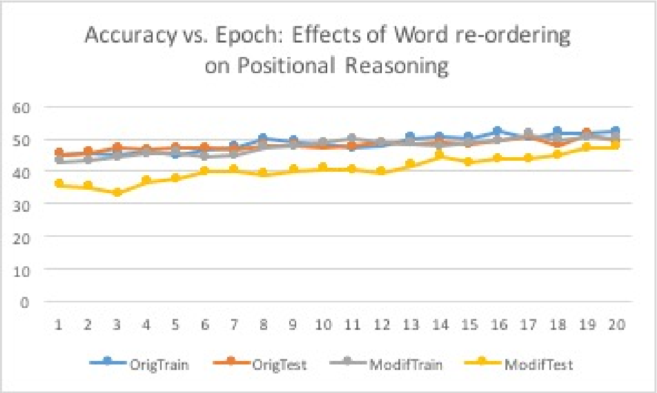


Figure 14: Changes in training and testing accuracy over 20 epochs in Positional Reasoning (Word re-ordering)

that word re-ordering did not have a drastic effect on the training accuracy of positional reasoning over 20 epochs - however, testing accuracy is slightly lower than it was originally, but it eventually approaches the testing accuracy value of the original data, as the number of epochs reaches 20. A reason for this lack of differentiation in converging accuracy between original and modified data could be that word-reordering changes the order of logic in Positional Reasoning or Path Finding, but it does not inherently change the logic itself. Hence, the training accuracy should be expected to follow closely even after modification because predictions to answers in questions in training data will likely not change much relatively to how the predictions were before. As for the testing accuracy, it is slightly unclear as to why the testing accuracy starts lower than it did in the original dataset testing. One hypothesis for why this is could be that the RNN initally overfits slightly to the modified training data, and then adjusts accordingly after more computation is performed through each epoch. Contrastingly figure 13 indicates a severely worse performance for both training and testing accuracy after word re-ordering, and this can be because the simple negation task is not a logical task, but a task with a standard subject-verb input (as are most other tier 1 tasks). Henceforth, once the episodic memory module analyzes much data of the same structural input, it could have difficulty recognizing occasional changes in structural input.

However, in positional reasoning and path finding (tier 2 tasks), the standard subject-verb input does not exist, and hence whatever accuracy would be lost in changed outputs is likely already lost (which is probably the reason that tier 2 tasks already have lower accuracy). The differences in training/testing accuracy percentages are averaged for before and after modification in table 2, and all difference measurements are positive as an absolute value change - no signs are needed since average accuracy before alterations was always higher than average accuracy after the modifications of tense shift or re-ordering were made. According to the 3rd, 4th, 7th, and 8th rows of table 2, it is evident that both tense shift and word re-ordering have higher impacts on decreased accuracy for simple negation than positional reasoning.

Contrast Categories (in terms of %, rounded to 3 significant figures)	**Training**	**Testing**
Average Accuracy Difference (Simple Neg Tense Shift)	7.01	27.1
Average Accuracy Difference (Positional Reas Tense Shift)	12.0	13.6
Normalized (Simple Neg Tense Shift) (×0.13)	0.911	3.52
Normalized (Positional Reas Tense Shift) (×0.20)	2.40	2.73
Average Accuracy Difference (Simple Neg Word Re-order)	35.4	38.6
Average Accuracy Difference (Positional Reas Word Re-order)	14.3	7.27
Normalized (Simple Neg Word Re-order) (×0.13)	4.60	5.02
Normalized (Positional Reas Re-order) (×0.20)	2.86	1.45

Table 2: Data on numerical alterations of accuracy percentage, averaged through 20 epochs and normalized according to degree of text file change in Simple Negation vs. Positional Reasoning

6.3. MCTest/Harry Potter Experimentation

Based on the implementation that was created for both MCTest and Harry Potter, with modifications including the addition of a choice selection module as well as the alteration of the episodic memory attention mechanism and answer module, figures 15 and 16 depict the training and testing accuracies of MC160 and MC500.

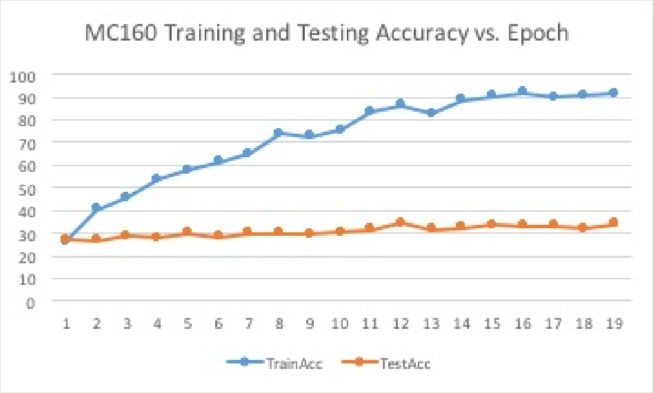


Figure 15: MCTest160 Accuracy Results

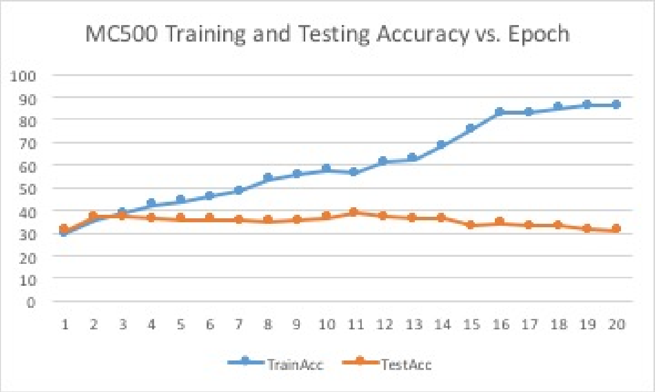


Figure 16: MCTest500 Accuracy Results

As indicated by the MC160 and MC500 plots, training accuracy slowly rises in each, but testing accuracy remains rather stagnant in MC160 and actually decreases from its original value in MC500. Perspectives on these outcomes shall be discussed in the summary section of this report. Accordingly, figures 17 and 18 depicts training and testing accuracies of the adjusted Harry Potter dataset for single sentence/paragraph granularity and multi-paragraph granularity. From figures 17 and 18, it is clear that training accuracy performs slightly better in Multi-paragraph passage data

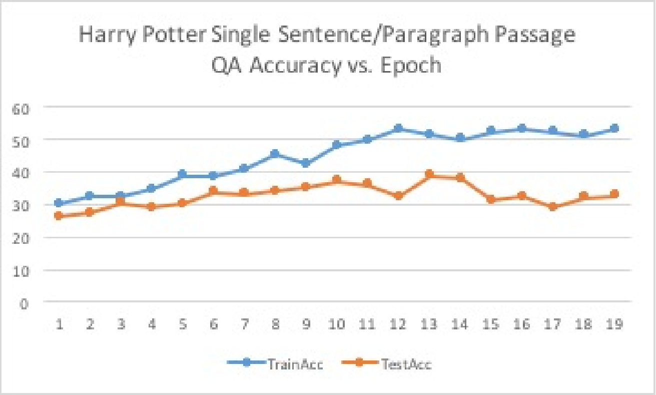


Figure 17: HP Single Sentence/Paragraph QA MC Accuracy

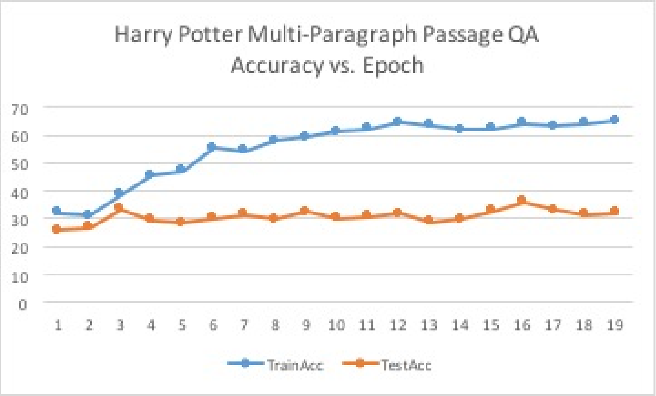


Figure 18: HP Multi-Paragraph QA MC Accuracy

than Single Sentence/Paragraph passage data, and the opposite is true for testing accuracy between the two data collections of differing length passages. A plausible reason for why the multi-paragraph Harry Potter model performs the way it does is that it has a similar length-structure to the original unmodified MC160 dataset, and the vocabulary has been ensured to be correspondingly within the range of MCTest. As for the single paragraph passage data, accuracy was slightly higher for testing, but lower for training than for the multi-paragraph data. The decrease in training accuracy could presumably be because less information could be justified to answer questions on a given passage, since the passage was simply shorter in length. As for the slightly increased performance in testing accuracy, a hypothesis for why this could be is that questions were generally simpler for these shorter passages, in that they needed less information backing, and there were more factual than inferential questions for the shorter passages.

7. Summary

7.1. Conclusions

The recently introduced Dynamic Memory Network (DMN), which is a neural network architecture that processes input sentence sequences and questions, instantiates episodic memories, and generates/identifies relevant answers, has a high potential for outperforming existing techniques in the area of Question-Answering in Natural Language Processing. However, past research within the DMN arena has dealt with very straightforward inputs consisting of simple-structure sentences which questions have been answered on (i.e. bAbI tasks). The goal of this paper was to introduce the concept of DMNs into answering questions about more complicated pieces of literature, for which the Harry Potter novels serve a starting point. Sentences from Harry Potter novels and the MCTest data set contain syntactic elements outside of subjects, verbs, direct objects, and prepositional phrases. More complex literature contains tense shifts, uncommon word orderings, more intricate types of phrases (such as appositive phrases), and many other syntactic elements that have not been currently explored within the textual data that DMNs have been created for in the past.

While past work has been done on DMNs, in terms of novelty, this paper introduces a new choice selection module that is helpful in expanding the repertoire from single-word simple-question answers to choice selection among a group of choices for more complex pieces of text (derived from Harry Potter novels and Microsoft's MCTest dataset). It also introduces structural modifications to the episodic memory module and answer modules, specifically to target more complex passages that questions are to be asked on. In addition to its functionality, it introduces the novel idea of making simple input data (i.e. bAbI) more intricate through syntactic adjustments such as tense shifting and word re-ordering, with the inherent goal of simulating the more complex sentence-based characteristics to be found in formal literature or more complex fiction. Within bAbI, it is discovered that both tense shifts and word-reordering have higher impact on testing accuracy of simple negation than positional reasoning, and since these two types of tasks serve as quintessences of tier 1 and tier 2 tasks respectively, it can be deduced that data alteration has less impact on tier 2 tasks, possibly because of their multi-relational nature. With respect to the DMN created for the MCTest dataset and the Harry Potter dataset, the top testing accuracy reached within a span of 20 epochs was 38.8 % for Harry Potter Single Sentence/Paragraph data, 35.9 % for Harry Potter Multiple Paragraph data, 34.2 % for MC160 data, and 38.6 % for MC500 data.

7.2. Project Limitations and Future Work

Certain limitations that were come across in the project were that, first of all, the MCTest dataset, and the Harry Potter dataset as well, are far too small for good enough training to bring about higher testing accuracy. For this reason, the plots indicate high increases in training accuracy, but nearly stagnant and sometimes even decreasing testing accuracy. The DMN model becomes highly accurate at predicting answers to passages it has already seen, but 660 stories is seemingly not enough data to answer machine-comprehension questions about fictional passages that have never been encountered before. Additionally, it is possible that the use of supervised training for the DMN also contributed to significantly higher training accuracies than testing ones. As for future work, it would be intriguing to explore, as mentioned before in the report, the possibility of automated validation via Pottermore (the Harry Potter encyclopedia), or any encyclopedia dedicated to a topic or fictional novel series. It is, without doubt, possible that human error occurred in my supervised answers to the training and testing data questions, especially since there were many. Furthermore, in the future, more QA data should be accumulated from various sources of literature, whether it be novels in a series, philosophical texts, historical texts, or more. A large part of my inability to achieve high testing accuracy within the Harry Potter dataset was that I had insufficient QA data to train on.

8. Acknowledgements

It was a true pleasure to be advised by Dr. Fellbaum this semester. Because of a hectic internship recruiting season, I was not able to present in seminar as many times as I had liked, but whenever I met with Dr. Fellbaum, she loved discussing my ideas and gave great advice. Many thanks to Dr. Fellbaum for her guidance and support in this project of mine.

9. Ethics

I pledge my honor that this project represents my work in accordance with University regulations.

Avinash Nayak

References

[1] Jonathan Su James Bradbury Robert English Brian Pierce Peter Ondruska Ishaan Gulrajani Ankit Kumar, Ozan Irsoy and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. arxiv preprint arxiv:1506.07285, 2015.

[2] 2016. https://github.com/YerevaNN/Dynamic-memory-networks-in-Theano YerevaNN, Dynamic memory networks in Theano. Github.

[3] Antoine Bordes Jason Weston, Sumit Chopra. Memory networks. arxiv preprint arxiv:1410.3916 (2014).

[4] Arushi Raghuvanshi and Patrick Chase. Dynamic memory networks for question-answering. stanford cs 224d, 2015.

[5] Qian Lin and Hongyu Xiong. Dynamic memory network on natural language question-answering.

[6] Wen-tau Yih Jianfeng Guo-Li Deng-Paul Smolensky Moontae Lee, Xiaodong He. Reasoning in vector space: An exploratory study of question answering. arxiv preprint arxiv:1511.06426v4, 2016.

[7] J.L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning, 7(2-3):195–225, 1991.*

[8] Richard Socher Jeffrey Pennington and Christopher D. Manning. 2014. glove: Global vectors for word representation.

[9] van Merrienboer B.-Gulcehre C. Bahdanau D.-Bougares F.-Schwenk H. Cho, K. and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. in emnlp, 2014.