Giving Machines Memory and Focus: Evaluating Attention-Based and Memory-Based Neural Network Models on Large Q&A Datasets

Dan Strawser

December 13, 2015

Abstract

This is what the project is about

1 Introduction

Question Answering tasks are some of the most general in natural language processing.

This project focuses on QA tasks where some type of attention and inference is required. For example, to answer the question posed in Fig. 1, the algorithm must be able to find the relevant phrases and infer Sally's motivation.

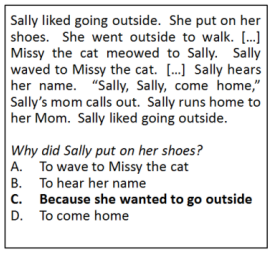

Figure 1: Example QA task from MC Test dataset. Taken from [1].

Many approaches have been proposed to solve this task, among them, feature-based methods. Neural networks are another approach to solving this problem. Recurrent neural networks seem especially well-suited for this problem because of their ability to use sequential data as input. More recent models such as LSTM (Long Short Term Memory) units and GRU (Gated Recurrent Units) make it possible to train on longer sequences of data by avoiding the problem of exploding gradients. However, some question and answer tasks require analyzing very large sequences of data and even these models have difficulty.

Advancements made in the past year in deep neural networks provide more promise. These networks combine the sequential encoding found in RNNs and add memory and attention. Attention mechanisms are important because, given a large amount of information and a question, finding the relevant information is a challenge. While these have shown promise on smaller corpora, it remains to be seen their advantages on larger datasets. The goal of this project is to evaluate these neural network architectures and compare them against one another. Specifically, I investigate the Memory Network described in [**?**] and the Dynamic Memory Network from [2].

2 Approaches

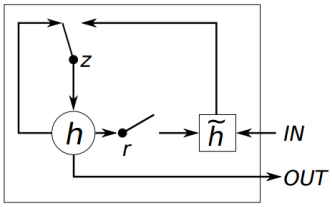

Figure 2: Gated Recurrent Layer Architecture. Taken from [3].

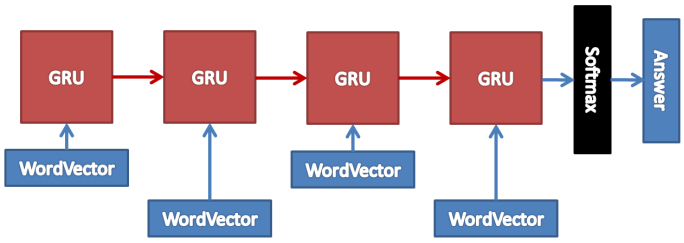

Figure 3: Simple GRU Encoder.

2.1 GRU Encoder

The baseline algorithm is simply a Gated Recurrent Unit (GRU) encoder, pictured in Fig. 2. This is simply a recurrent neural network that reads in each word of a sequence and, through a softmax layer, produces an output answer. To form the sequence, a vector representing the question is concatenated with a vector containing the article (or information from which an answer is generated). While the inputs can be generated by Word2Vec, for these experiments they were either one-hot vectors or word indices.

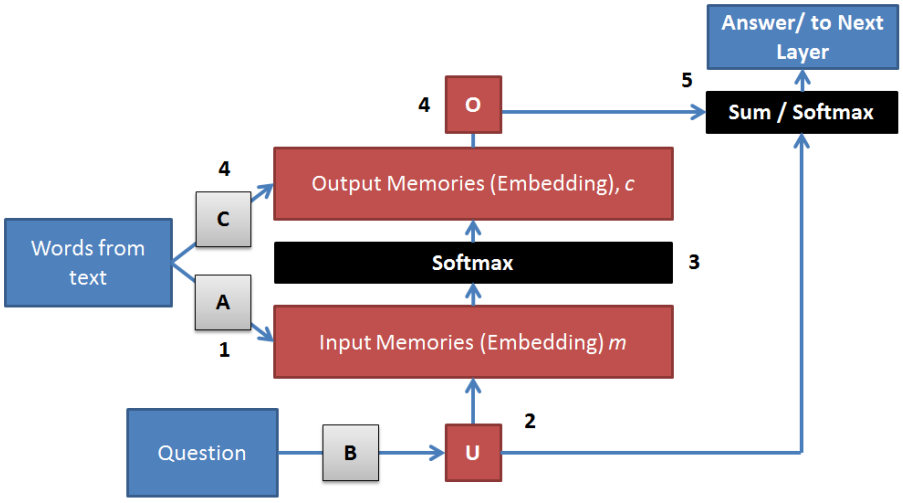

Figure 4: Memory Network Overview.

The base unit for the Encoder (and also the subsequent Dynamic Memory Network) is the Gated Recurrent Unit. The unit consists of two gates, a reset gate and an update gate, which determine whether or not to update the unit's hidden state with an input or simply propagate the previous hidden state.

2.2 Memory Network

More recently, the Memory Network was described in [4] as an End-to-End algorithm. This is important because it means that the algorithm only requires (*text*, *question*, *answer*) tuples instead of relying on fact annotations. The memory network is relatively simple in the fact that is relies on "memory" embeddings and softmax attention mechanisms.

For example, as described in [4], an input sequence of words W and an input question Q are both transformed into embeddings through embedding matrices A and B respectively. Question embeddings are transformed into "memory" embeddings m (Numbered **1** in Fig. 4) and questions into embeddings b (Numbered **2** in Fig. 4). At this point a softmax layer is applied between the input and question embeddings (Numbered **3** in Fig. 4):

$$p_i = \text{Softmax}\left(u^T m_i\right) \tag{1}$$

Intuitively, this softmax determines which memories are relevant to answering the question. In addition to the memory embeddings m_i, the input stream is also embed into an output embedding, C. The result of the softmax layer, Eq. 1, is multiplied by this output embedding, Numbered **4**. This result passes through a summation with the question state u, numbered **5**. The result of the

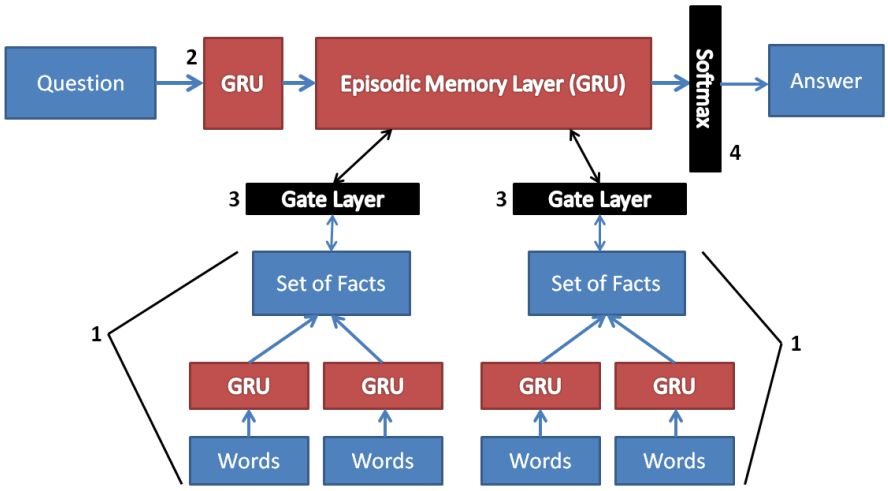

Figure 5: Dynamic Memory Network Overview. The DMN shown represents a DMN that reads a document twice.

sum at **5** in Fig. 4 can either be passed to a softmax for a final answer or passed to another memory network layer. That is, the layers are stacked and the result of one is passed to another as an input state u.

2.3 Dynamic Memory Network

The final model investigated is the Dynamic Memory Network as presented in [2]. The dynamic memory network is somewhat similar to the Memory Network; however, it relies more explicitly on GRU encodings and adds the potential of "re-reading" sentences.

First, each sentence is encoded into a fact through a GRU-based RNN. This RNN reads each word in the sentence and updates its hidden state. The final output, a "fact" encoding, is passed to the next layer. This process is repeated for the entire document to generate a set of fact encodings, which is shown as **1** in Fig. 5.

Next, given a set of facts from the entire document, it must be determined which of them is relevant to answering the given question. For this, a gating mechanism is deployed, numbered **3**. The functional form of this gate described in [2] is:

$$z\left(c_i, m, q\right) = \left[c_i, m, q, c_i \circ q, c_i \circ m, |c_i - q|, |c_i - m|, c_i^T W^b q, c_i^T W^b m\right] \quad (2)$$

$$g_i\left(c_i, m, q\right) = \sigma\left(W^a \tanh\left(W^c z\left(c_i, m, q\right)\right)\right) \quad (3)$$

where c_i is the i^{th} fact encoding, m is the hidden state of the Episodic Layer (described below), and q is the question encoding. The magnitude of gate g_i

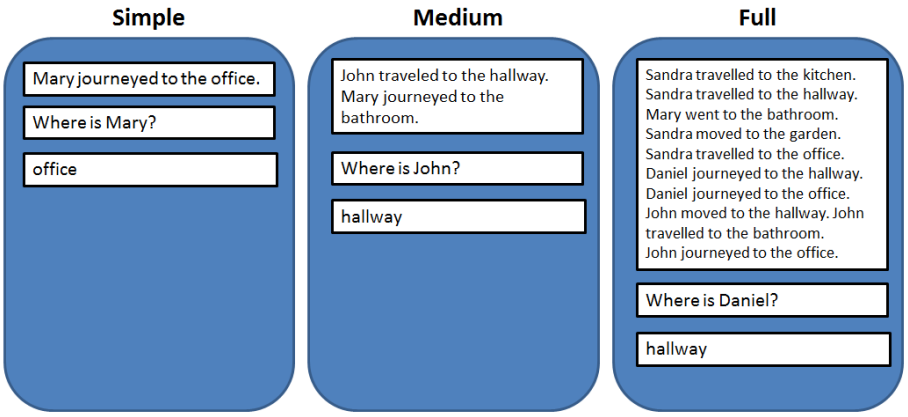

Figure 6: Examples of Babi tasks used for this project.

- a scalar - determines whether or not fact encoding c_i will be passed to the episodic GRU-RNN. This is done through a update equation:

$$h_t = 1 = g_t \text{GRU}\left(c_t, h_{t-1}\right) + \left(1 - g_t\right) h_{t-1} \tag{4}$$

The top episodic layer is used to read the facts and produce a final output.

One of the advantages of the DMN is that it can re-read a series of episodes. This may be advantageous when the ordering of facts matters but when the second fact in the text occurred *before* the first fact in the text, i.e. *John went to the store on Saturday. On the Wednesday prior, John drove to California.*. Therefore, the example shown in Fig. 5 is shown reading a document twice, producing two sets of facts, choosing episodes for each of these and then, finally, producing an output answer.

3 Datasets

One of the challenges with QA tasks is designing datasets that are nontrivial and require inference to answer questions but are also large enough that the neural network will learn well.

3.1 Babi Tasks

The Babi Tasks are a set of 20 tasks created by researchers at Facebook and are described in . The motivation is to create a set of tasks to test a general question answering algorithm. The algorithm should be general in that it should be able to perform well on all of them and not just a few. The set contains a wide variety of tasks. For example, understanding orientation: *The hallway is east of the bathroom. The bedroom is west of the bathroom. What is the bathroom east of?* or understanding counting objects: *Mary moved to the bathroom. John*

went to the kitchen. Mary took the football there. How many object is Mary carrying?.

For this project, I focused on Babi Task 1, which involves determining where people went. I split the task into simple, medium, and full versions as pictured in 6. The simple and medium were primarily for debugging - even the simple GRU encoder can easily obtain 100 % correct responses on the simple version.

An advantage of the Babi set is that it tests the neural networks on attention and inference. However, their corpora is limited in size (Babi Task 1 only contains about 20 different words) and does not represent actual corpora that one may see in the real world.

3.2 WikiQA

The second dataset that I used was one suggested by researchers at Carnegie Mellon involving Wikipedia article and questions that can be answered upon reading the article [5]. One example is shown in 7. An advantage of this dataset is that it represents a more realistic corpus size than the Babi Tasks. However, a major disadvantage of the set is that it is very small. It consists of approximately 200 articles and 5,000 questions concerning those articles. As seen in the results, the neural networks have difficulty learning anything but basic yes/no questions on this dataset.

3.3 Google - CNN

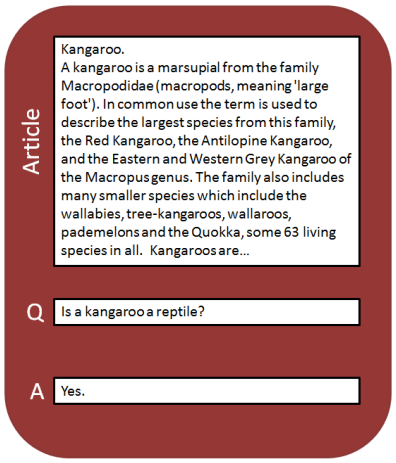

Figure 7: Example of article, question, and answer tuple from WikiQA dataset.

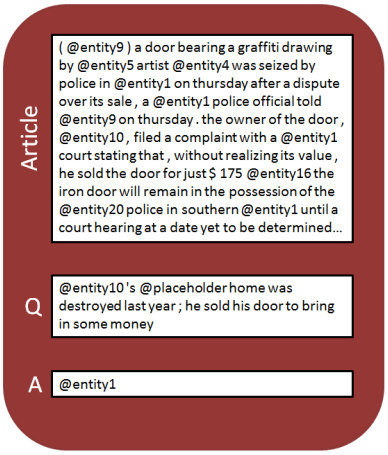

Figure 8: Example of Google-CNN article, question statement, and answer tuple.

The third dataset that I considered was provided by researchers at Google

and described in [6]. This dataset consists of articles from CNN (another similar dataset from Daily Mail is also described) with autonomously generated question statements and answers. The question statements are generated through the simple, but effective, observation that all CNN/Daily Mail articles have accompanying human-written summaries. Words can be replaced from these summaries to create question statements where an algorithm attempts to predict the word that was removed. The advantage with creating training examples autonomously from a source like CNN is that a huge number can be produced (the full dataset is approximately one million articles). While the WikiQA dataset has the problem of too few examples, this is not an issue with the CNN dataset.

References

[1] K. Narasimhan and R. Barzilay, "Machine comprehension with discourse relations," *ACL 2015*, 2015.

[2] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," *CoRR*, 2015.

[3] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014.

[4] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," *NIPS 2015*, 2015.

[5] N. A. Smith, M. Heilman, and R. Hwa, "Question generation as a competitive undergraduate course project," *Proceedings of the NSF Workshop on the Question Generation Shared Task and Evaluation Challenge*, 2008.

[6] K. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," *NIPS 2015*, 2015.

[7] P. Kapashi, Darshan; Shah, "Answering reading comprehension using memory networks," *Stanford 224d Course Project*, 2015.

[8] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *NIPS 2014*, 2015.

[9] J. Weston, A. Bordes, S. Chopra, T. Mikolov, A. M. Rush, and B. van Merrienboer, "Towards ai-complete question answering: A set of prerequisite toy tasks," *arXiv:1502.05698*, 2015.