Question Answering Model

Introduction

Question answering is a highly intricate and demanding natural language processing (NLP) task. It necessitates a nuanced understanding of sentence context and underlying emotions to formulate precise query responses. Questions may directly relate to the contextual details of a sentence or require insight into the deeper essence upon which the model is trained.

The question-answering module is a testament to its versatility, as it undertakes various tasks. For instance, machine translation accurately translates given text into French, a feat that demands a keen grasp of both literal meaning and nuanced expression. Its capabilities extend to named entity recognition, allowing for identifying and classifying critical pieces of information within a text, such as individuals, locations, and organizations, according to predefined categories. Moreover, the module's proficiency in parsing for parts of speech enhances its understanding of language syntax and semantics, leading to more accurate and contextually relevant responses to queries.

In essence, the question-answering model represents a significant advancement in natural language processing, offering a diverse range of capabilities from machine translation to named entity recognition and parts of speech tagging. As these models continue to evolve, the potential of their applications and benefits are limited only by our imagination.

Our model leverages the Dynamic Memory representation, incorporating an attention mechanism to fulfil its task. Through training on the Babel dataset, which comprises input sentences, questions, and correct answers, the model acquires the ability to generate answers, subsequently subject to rigorous evaluation for accuracy.

The Dynamic Memory Network (DMN) commences by computing representations for both the input sentences and the posed question. Following this, the question representation triggers an intricate iterative attention process probing the inputs to retrieve pertinent facts. These retrieved facts are then processed by the DMN memory module, culminating in providing a comprehensive vector representation embodying crucial information to an answer module responsible for crafting the final answer.

This project aims to develop a model capable of accurately responding to questions. This model should excel in two main areas: firstly, understanding and addressing queries directly tied to the context of the provided text, and secondly, interpreting and responding to questions that pertain to the emotional tone of the sentence. The model aims to achieve a comprehensive understanding and versatile response capability through this dual-focus approach, enhancing its effectiveness in processing and interacting based on textual information.

Methodology

The document delves into using the Dynamic Memory Network (DMN) to respond to queries pertinent to a given text. The model comprises four core segments: the input module, question module, episodic memory module, and answer module.

Input Module:
The input module processes and encodes the raw sentences provided to the network, transforming them into a distributed vector representation. It handles various input types, including articles, news pieces, and diverse textual content.

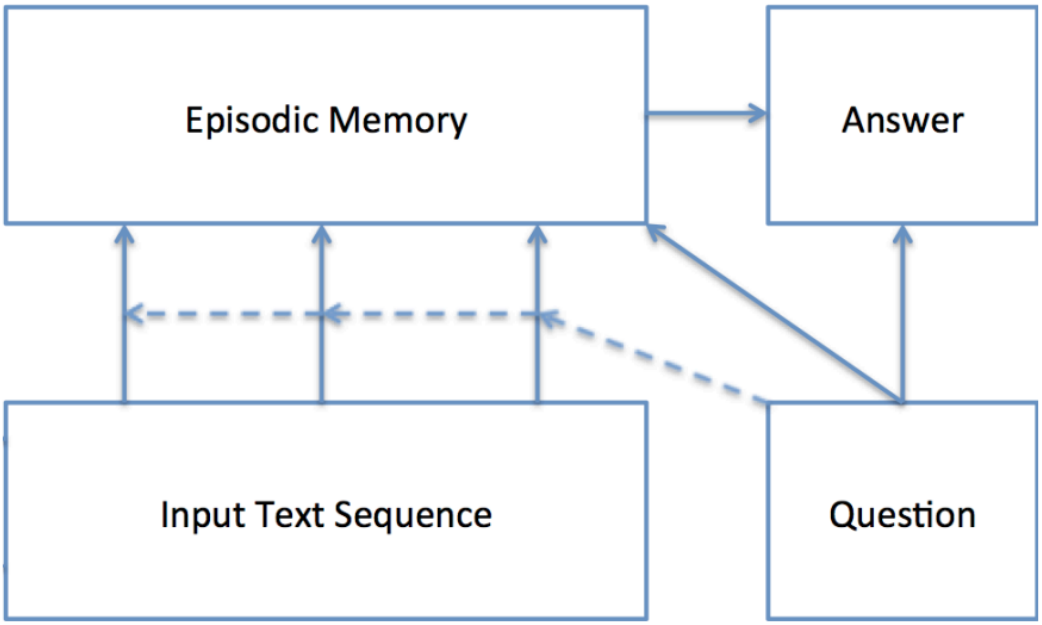
Question Module:
Like the input module, the question module encodes the textual content of the question and generates its vector representation.

Episodic Memory Module:
Upon obtaining the input and question representations, this module processes and distils the context that requires emphasis. Subsequently, it produces a memory representation by synthesizing the question and context representations. The resulting output is then directed to the answer module.

Answer Module:
Utilizing the memory representation from the episodic memory module, this module addresses queries within the provided dataset.

Model architecture

As mentioned above the architure of the model has total 4 parts are the functionality of the model is also mentioned in this we will look into it briefly.



Input Module
In natural language processing tasks, the input comprises a sequence of words represented as w1, w2, ..., wTI. One commonly used method to capture this input sequence is through the use of a recurrent neural network (RNN). In this approach, the RNN takes word embeddings as inputs and updates its hidden state at each time step t based on the formula ht = RNN(L[wt], ht-1), where L is the embedding matrix and wt is the word index of the tth word in the input sequence.

When the input sequence represents a single sentence, the input module produces the hidden states of the recurrent network. However, if the input sequence consists of a list of sentences, we aggregate the sentences into a long list of word tokens and introduce an end-of-sentence token after each sentence. The hidden states at each end-of-sentence token then serve as the final representations of the input module. In subsequent sections, we denote the output of the input module as the sequence of TC fact representations c, with ct representing the tth element in the output sequence.

In the scenario of a single sentence input, TC equals TI, implying that the number of output representations is equivalent to the number of words in the sentence. For an input list of sentences, TC corresponds to the number of sentences.

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

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

$$\tilde{h}_t = \tanh\left(Wx_t + r_t \circ Uh_{t-1} + b^{(h)}\right)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t$$

Question Module

Same concept is applied for the question that we have used in the input module.
The Episodic Memory Module is ingeniously designed to process and iterate over input representations, continually refining its internal memory storage. At the core of its operation is a sophisticated duo: an attention mechanism paired with a recurrent network. This pairing allows the module to meticulously update its memory at each iteration.

Imagine the attention mechanism as a spotlight, focusing intently on different pieces of input information, known as fact representations `c`. It doesn't work in isolation. Instead, it draws on the context of a specific question representation `q`, along with the insights gathered from the previous memory state `mi-1`. By harmonizing these elements, it crafts a distinct episode `ei`. This freshly minted episode is not merely stored; it is actively used, in tandem with the reservoir of past memories, to update the episodic memory to a new state, denoted as `mi`. This update process employs a Gated Recurrent Unit (GRU), with the initial state `m0` ingeniously set to the question vector `q` itself, embedding the question at the heart of memory evolution.

For certain tasks, the complexity and depth of the information necessitate not just a single overview but multiple passes over the input data. This repetitive process ensures a richer, more nuanced understanding, culminating in the final memory state `mTM` after `TM` passes, which is then poised to inform the answer module.

The module intricately applies a tailored GRU mechanism to sift through and update memories with a sequence of inputs labeled as `c1`, `c2`, ..., `cTC`. Unlike a one-size-fits-all approach, it discriminately processes these inputs through a series of gates, known as `gi`,

which meticulously assess and determine their significance in the memory refinement process. This thoughtful processing culminates in the creation of an episode vector, a richly detailed composite that captures the essence of the episodic memory's interactions with the input data. This vector then serves a critical role in the answer module, empowering it to craft responses or draw conclusions from the intricate web of information woven into the episodic memory.

This architecture is a testament to a meticulous blend of precision, recurrent learning capabilities, and the flexibility to assimilate and evolve with a continuous influx of information, all while being steered by the pivotal question that propels the inquiry forward.

$$h_t^i = g_t^i GRU(c_t, h_{t-1}^i) + (1 - g_t^i) h_{t-1}^i$$
$$e^i = h_{T_C}^i$$

Stopping Criteria: Within the episodic memory system, there's a built-in mechanism to halt the review of inputs. This is accomplished by integrating a special marker, dubbed the 'end-of-passes' marker, into the input stream, effectively signalling the system to cease the iterative focus process should this marker get selected through the gate function. In situations where there's a lack of direct oversight on datasets, we establish a cap on the number of iterations to ensure a finite stopping point. This design ensures that the module operates with end-to-end differentiability, promoting a seamless learning experience.

The Answer Module: Tailored to the task at hand, the answer module springs into action, tasked with generating a response based on a provided vector. This activation occurs either once at the culminating point of the episodic memory journey or iteratively at every step of the process. Center stage in this module is an additional GRU (Gated Recurrent Unit), with its initial state cleverly initialized to the last memory slice (a0 = mTM). Each step of the way, this GRU considers a concoction of inputs including the question (q), the preceding hidden state (at-1), and the prior predicted output (yt-1), weaving them into its calculations. The concoction includes a concatenation of the last generated word alongside the question vector, thereby enriching the input at every timestep. Moreover, the outputs undergo training through a cross-entropy error classification mechanism, finely tuned to recognize the correct sequence augmented with a distinctive end-of-sequence token.

In scenarios where the task is to model a sequence, the ambition is to accurately tag each word within the original sequence. To this aim, the DMN (Dynamic Memory Network) adheres to a similar protocol as articulated above, save for one adjustment - Equation 8 metamorphoses to accommodate each word's unique identity (ei = hi t). A noteworthy mention is the efficiency garnered from the gates' uniformity during the initial pass, attributed to the constant nature of the question. This singularity allows for a computational speed-up by negating the need for repetitive gate computations. However, as the narrative unfolds across different episodes, the gates adapt and diverge, reflecting the evolving context.

Training Paradigm: The training regimen casts itself as a supervised classification challenge, with the primary objective to minimize the cross-entropy error associated with the answer sequence. For datasets enriched with gate supervision, like the bAbI dataset, the error metrics of the gates are assimilated into the overarching cost structure. What sets this model apart is its collaborative infrastructure, where all modules interlace through vector representations, supported by an array of differentiable and deep neural networks outfitted with gates. This intricate network architecture lends itself to being honed through the principles of backpropagation and gradient descent, ensuring a robust and adaptive learning framework.

Question Answering
The Facebook bAbI dataset serves as a synthetic benchmark for evaluating a model's ability to comprehend and reason based on provided facts. It presents various tasks that assess different skills essential for a question-answering model, including coreference resolution, deduction, and induction. While successful performance on this dataset is a crucial requirement, it should be noted that it does not guarantee similar proficiency when applied to real-world text data.

During training on the bAbI dataset, the objective function is defined as J = ECE(Gates) + ECE(Answers), where ECE represents the standard cross-entropy cost and the parameters are adjusted using hyperparameters. In practical terms, training commences with one parameter set to 1 and the other set to 0. Subsequently, the first parameter is switched to 1 while keeping the second parameter at 1.

In Section 2.1 of the dataset, the input module extracts fact representations by utilizing the encoder's hidden states at time steps corresponding to the end-of-sentence tokens. The gate supervision aims to select one sentence per pass, leading to an exploration of modifying Eq. 8 to a simple softmax instead of a GRU. This alteration has resulted in improved outcomes, possibly due to the softmax's ability to encourage sparsity and effectively select one sentence at a time.

Result

```
TEXT:  mary travelled to the bedroom . mary journeyed to the bathroom . mary got the football there . mary passed the football to fred .
QUESTION:  who received the football ?
RESPONSE:  mary Incorrect
EXPECTED:  fred

TEXT:  bill grabbed the apple there . bill got the football there . jeff journeyed to the bathroom . bill handed the apple to jeff . jeff handed the app
QUESTION:  what did bill give to jeff ?
RESPONSE:  apple Correct
EXPECTED:  apple

TEXT:  bill moved to the bathroom . mary went to the garden . mary picked up the apple there . bill moved to the kitchen . mary left the apple there . j
QUESTION:  what did jeff give to fred ?
RESPONSE:  apple Incorrect
EXPECTED:  football

TEXT:  jeff travelled to the bathroom . bill journeyed to the bedroom . jeff journeyed to the hallway . bill took the milk there . bill discarded the mi
QUESTION:  who gave the milk to bill ?
RESPONSE:  jeff Incorrect
EXPECTED:  mary

TEXT:  fred travelled to the bathroom . jeff went to the bathroom . mary went back to the bathroom . fred went back to the bedroom . fred moved to the c
QUESTION:  who received the football ?
RESPONSE:  mary Incorrect
```