E1 246 - Natural Language Understanding
Final Report: Question Answering

Sachin Mittal, Nidhi Kumari
sachinmittal,nidhikumari@iisc.ac.in

Abstract

Several deep learning models have been proposed for question answering. In this project we studied the application of various deep learning models to the question answering task. After describing some of them in this report, we focus our attention on DMN+, which is an improvement to DMN and have provided state-of-the-art results on some QA tasks while being relatively fast to train.

1 Introduction

Question Answering (QA) is one of the oldest tasks in NLP. Most problems in NLP can be formulated as a question answering task, and QA has recently seen commercial popularity and media attention in applications such as Siri and Watson.

Original QA systems often involved developing a structured knowledge database that is handwritten by experts in a specific domain. In these systems, a question asked in natural language must be parsed and converted into a machine-understandable query that returns the appropriate answer.

With the massive amounts of natural language information on the web, current systems focus on extracting information from these documents. As a result, recent QA systems focus on information-retrieval based methods which include 1) a question processing module for formulating a query, 2) an information-retrieval module for selecting the appropriate document and passage, and 3) an answer processing module to generate the appropriate answer in suitable language. Many of these applications are open-domain, meaning they can answer questions about any topic.

Recently with advancements in deep learning, papers have been published that utilize recurrent neural networks for question answering. These deep networks generate latent representations of natural language text passages rather than relying on extracted features such as part of speech tagging, parsing, named entity recognition, etc. These networks require much less pre-prossesing and have recently matched and even exceeded the results of other models.

We approach the question answering task using the DMN+ model. We implemented DMN+ in Pytorch and train and test the model on the dataset described below.

2 Related Work

Prior to DMN+, work had been done in the related lines of attention and memory mechanisms. Prior to DMNs, work had been done in the related lines of attention and memory mechanisms. Wetson et al [7] first presented memory networks as a way to use a long-term memory component as a dynamic knowledge base for question answering. This memory network, unlike DMN+, requires the labeled supporting facts during training. Attention mechanisms have recently been used for a variety of applications including image captioning [8]. Stollenga et al proposed a model that allows the network to iteratively focus its internal attention on some of its convolutional filters. Similarly, DMNs use attention for QA to iteratively focus on certain sentences in the input text.

There have been a few papers published within the recent year that have presented Dynamic Memory Networks and improvements on the model. DMN+ gained popularity with Xiong et als [1] . They present the DMN+ model described below and apply it to a variety of language tasks including the Facebook bAbI dataset.

3 Motivation

The advent of the Internet has resulted in a massive information explosion. We need to have an effective and efficient means of locating just the desired information. In search engines like Google, When a user asks a question about a specific entity, Google reads the corresponding text (usually a Wikipedia article) and attempts to locate the answer within the document.

This type of systems can additionally allow a company to search through client data, a doctor to search through medical records, or a movie goer to ask for a recommendation. Solving the problem efficiently and accurately has the potential to improve these tasks and the many others like them. QA has application in a wide variety of tasks, such as information retrieval and entity extraction. Recently, QA has also been used to develop dialog systems and chatbots designed to simulate human conversation.

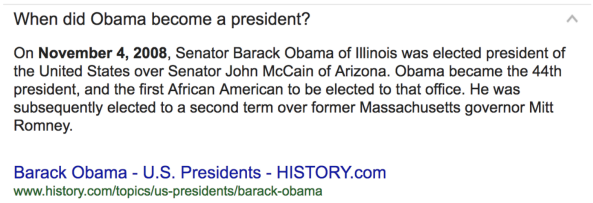

Figure 1: : Google Search OneBox demonstrating the question answering task

4 Literature Review

Here we are summarizing three main papers. DCN [2] and R-NET [3] is used for SQuAD dataset [4] and DMN+ [1] is used for fb bAbI 10k dataset [5]

1. **DCN**
   DCN is used in SQuAD kinds of dataset where answer spans in paragraph itself, The goal of this model is to predict an answer span tuple $\{a_s, a_e\}$, where a_s is the index into the context paragraph of the first token in the answer, and a_e is the index of the last token in the answer. DCN iterates over potential answer spans and this way it does not get stuck in local maxima. (Techniques which uses single pass nature have no way to recover from local maxima corresponding to incorrect answers). The model consists of a co-attention encoder and a dynamic decoder.

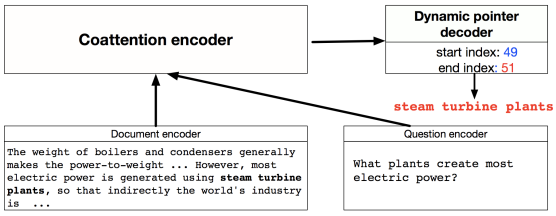

Figure 2: : Overview of the Dynamic Coattention Network

   1.1 The encoder first encodes the question and the document separately, then builds a codependent representation through co-attention.

      (a) Document and Question encoder: These embeddings are computed with the same LSTM. If document has m words and question has n words then after encoding we have - $D = [d_1, d_2, \ldots d_m]$, $Q = [q_1, q_2, \ldots q_n]$

      (b) Coattention encoder:
      Now DCN performs coattention mechanism that attends to the question and document simultaneously, and finally fuses both attention contexts.
      It first compute the affinity matrix, which contains affinity scores corresponding to all pairs of document words and question words, $L = D^T Q$. The affinity matrix is normalized row-wise and column-wise to produce the attention across the document and question, A^Q and A^D. And then encoder produces codependent representation through co-attention.

   1.2 Dynamic Pointer decoder
   The decoder then produces a start and end point estimate given the co-attention. Decoder iteratively hypothesizing answers until the hypothesis no longer changes i.e. It halts when both the estimate of the start position and the estimate of the end position no

longer change, or when a maximum number of iterations is reached.

2. **R-NET**
   The architecture of R-NET is designed to take the question and the passage as inputs and to output an interval on the passage that contains the answer. The process consists of several steps:

   (a) Encode the question and the passage:
   This module encodes both question and the passage, Each word is represented by a concatenation of two vectors: its GloVe vector and another vector that holds character level information. To obtain character level embeddings it uses an Embedding layer followed by a Bidirectional GRU cell wrapped inside a TimeDistributed layer. Basically, each character is embedded in H dimensional space, and a BiGRU runs over those embeddings to produce a vector for the word. The process is repeated for all the words using TimeDistributed layer.

   (b) Obtain question aware representation for the passage
   The next module computes another representation for the passage by taking into account the words inside the question sentence.

   (c) Apply self-matching attention on the passage to get its final representation
   The output of the previous step (Question attention) represents the encoding of the passage while taking into account the question. The authors argue that the vectors have very limited information about the context. Self-matching attention module attempts to augment the passage vectors by information from other relevant parts of the passage.

   (d) Predict the interval which contains the answer of the question
   Finally it is ready to predict the interval of the passage which contains the answer of the question. To do this it uses Question-Pooling layer followed by Pointer-GRU.
   Question-Pooling is the attention pooling of the whole question vector . Its purpose is to create the first hidden state of Pointer-GRU.
   Pointer-GRU is a recurrent network that works for just two steps. The first step predicts the first word of the answer span, and the second step predicts the last word.

3. **DMN+**
   The dynamic memory network (DMN) +is one example of a neural network model that has both a memory component and an attention mechanism.

   At a high level, there are four modules, all of which communicate with vectors, and can be trained with backpropagation.

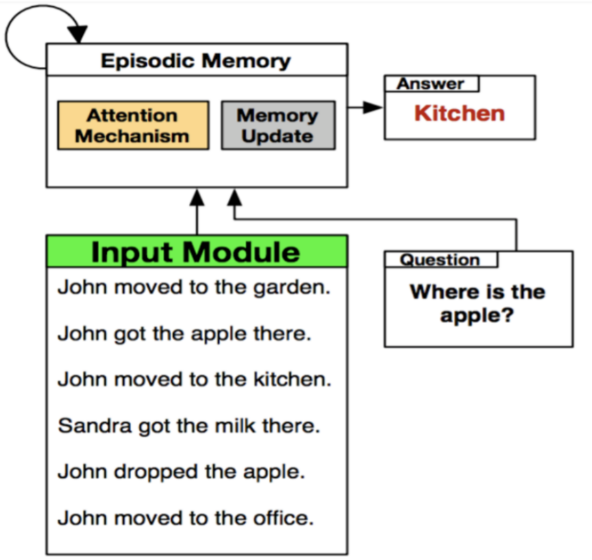

Figure 3: :Dynamic Memory Network

3.1 Input Module: This module consists two parts

   (a) Positional Encoding:
   Initially each fact is represented by list of words $f_i = [w_1, \ldots, w_{M_i}]$, here each word is represented using word embedding and After that weighted sum of all words is taken for each sentence and weight is defined by vector l.

$f_i = \sum_M^{j=1} l_j \circ w_j^i$ where l is $l_{jd} = (1 - j/M) - (d/D)(1 - 2j/M)$.

(b) The input fusion layer:
The input fusion layer takes these input facts and enables an information exchange between them by applying a bi-directional GRU.
$\overrightarrow{f_i} = GRU_{fwd}(f_i, \overrightarrow{f_{i-1}})$
$\overleftarrow{f_i} = GRU_{bwd}(f_i, \overleftarrow{f_{i+1}})$
$\overleftrightarrow{f_i} = \overleftarrow{f_i} + \overrightarrow{f_i}$
This allows contextual information from both future and past facts to impact $\overleftrightarrow{f_i}$

3.2 Question module: Now this will compute a vector representation q for question, which is simply just the final hidden state from a GRU where it feeds in the words in the question.

3.3 Episodic memory module:

The episodic memory module has two components: attention mechanism and memory update mechanism. The **attention mechanism** results in a context vector c that offers relevant information of the input at pass t. The inputs are the input facts F, question vector q and the previous memory m^{t-1}. Then the **memory update** mechanism uses the context vector c and the previous memory m^{t-1} to create m^t. The final pass T will result in memory m^T which has all the information needed to answer question q.

Because some questions require multiple passes over the memory, this module allows attention to be applied to the input multiple times and produces a final vector representation for all relevant inputs.

3.4 Answer Module: After multiple episode passes, we get the final M^T, which gets passed to the answer module.
$a = [q; m^T]$
$y^t = softmax(W(a))$ (y_t is the vector of vocab size having probability of each word being answer.)

5 Dataset description

Primarily our focus is on two datasets- fb bAbI and SQuAD

fb bAbI 10k - The Facebook bAbI-10k dataset has been used as a benchmark in many question answering papers. It consists of 20 tasks. Each task has a different type of question such as single supporting fact questions, two supporting fact questions, yes no questions, counting questions, etc. The input is a variable length passage of text. The type of question and answer depends on the task. For example, some tasks have yes/no answers while others are focused on positional reasoning or counting. Every answer in the bAbI dataset is one word, as evaluation is then clear-cut, and is measured simply as right or wrong. All of the tasks are noiseless and a human able to read that language can potentially achieve 100 % accuracy. The bAbI tasks cover far more than trivial comprehension however - they're supposed to represent a prerequisite towards an AI-Complete question answering solution. Each task aims to require a unique aspect of text and reasoning, testing the different capabilities of the learning models. To answer the questions correctly, the models must be able to perform induction, deduction, fact chaining, and more.

SQuAD - The reading passages in SQuAD are from high-quality wikipedia articles, and cover a diverse range of topics across a variety of domains, from music celebrities to abstract concepts. A passage is a paragraph from an article, and is variable in length. Each passage in SQuAD has accompanying reading comprehension questions. These questions are based on the content of the passage and can be answered by reading through the passage. One defining and important characteristic of SQuAD is that the answers to all of the questions are segments of text, or spans, in the passage i.e. dataset constrains the answer to be a continuous sub-span of the provided passage. The questions and answers in SQuAD were produced by crowdsourced humans, which makes them more realistic. The neural network needs to understand both the passage

and the question in order to be able to give a valid answer.

6 Model description

After Literature Review, We found out some existing techniques to solve QA problem. And implemented DMN+ (for text QA) that is already performing well on fb bAbI 10k dataset.

For the one word answers in the bAbI dataset, we frame the problem as a multi-class classificaion problem, and use a softmax categorical cross-entropy loss function. We can then evaluate the model by calculating the accuracy on the test set.

7 Result

Task ID	git repo	DMN+ paper
1	100%	100%
2	96.8%	99.7%
3	89.2%	98.9%
4	100%	100%
5	99.5%	99.5%
6	100%	100%
7	97.8%	97.6%
8	100%	100%
9	100%	100%
10	100%	100%
11	100%	100%
12	100%	100%
13	100%	100%
14	99%	99.8%
15	100%	100%
16	51.6%	54.7%
17	86.4%	95.8%
18	97.9%	97.9%
19	99.7%	100%
20	100%	100%

8 Proposed Experiment (Future Work)

Encodings for question and passage are independent in DMN+, It is good idea to explore encoding technique ideas from DCN which uses codependent representations for both questions and Passage.

It is also unclear what is the correlation between the number of passes when doing the memory update and the final accuracy. It might be valuable to experiment and discuss on how the number of passes would affect testing accuracy.

The bAbI dataset was useful in verifying that our implementation was correct. However, as a synthetic dataset, it may not accurately represent some of the difficulties of training on human generated dataset. It would be interesting to see how DMNs perform on more diverse datasets such as the DeepMind reading comprehension dataset [6].

9 Github Link

https://github.com/SachinMittal28/NLUFinalProject

Refrences

[1] Xiong, Richard et al. (2016) Ask Me Anything: Dynamic Memory Networks for Visual and Textual Question Answering. URL: https://arxiv.org/abs/1603.01417

[2] Richard Socher et al. (2018) Dynamic Coattention Networks For Question Answering URL: https://arxiv.org/abs/1611.01604

[3] Natural Language Computing Group, Microsoft Research Asia (2017) R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS URL: https://www.microsoft.com/en-us/research/publication/mrc/

[4] Rajpurkar et al. (2016) SQuAD: 100,000+ Questions for Machine Comprehension of Text URL: https://arxiv.org/abs/1606.05250

[5] Weston et al. (2015) TOWARDS AI-COMPLETE QUESTION ANSWERING: A SET OF PREREQUISITE TOY TASKS URL: https://arxiv.org/abs/1502.05698

[6] Hermann, Karl Moritz et al. (2015) Teaching Machines to Read and Comprehend. URL: https://arxiv.org/abs/1506.03340

[7] Weston et al.(2015) Memory Networks URL:https://arxiv.org/abs/1410.3916