Machine Reading Comprehension

Prasanth Sikakollu, Shweta Kedas, Uttam Singh
Department Of Computer Science And Engineering
National Institute Of Technology, Rourkela
Rourkela, India

***Abstract*—Reading paragraphs, understanding the related questions and answering them has always been a difficult task for the machines. It is only the capability of humans to understand the logic and meaning of the question and answer it to maintain a proper mode of interaction. But for machines, this is a quite complex task. The main focus of this paper is to explore the various machine learning and neural networks based techniques to develop and train a model on context of paragraphs, related questions and answers and then test the model on an user given paragraph and question. This paper will give a thorough understanding of data analysis of the SQuAD dataset and word embedding applied on the questions and answers of training set of the SQuAD dataset.**

I. Introduction

Machine Reading Comprehension is a task in both natural language processing and artificial intelligent research. The goal is to train a machine to understand a given passage and then answer the questions related to the passage. As comprehension models have gone mainstream, the expectations from such models have grown to expect human like abilities to answer the queries. Reading comprehension is a task that requires question answer system to read a text, process it, comprehend and be able to extract the span of the text and answer to the query.

Machine Reading Comprehension enables computer system to read a paragraph and answer some questions against it. While this is much easier task for humans, its not quite straight forward for machines. Machine tries to capture nested context in a paragraph, i.e., the subject, the predicate, the concept and the conditions from the question.

The model which we are developing uses all these captured aspects and puts them together to arrive at the answer. The model utilizes word embeddings, a technique in Natural Language Processing (NLP), Bidirectional Recurrent Neural Networks (BiRNN) and utilizes the attention in neural networks to highlight some part of the text under the context of the other.

We will be using and exploring the SQuAD dataset and develop a MRC model which could be trained on the dataset.

To understand the the working of our model let us first understand the basic difference in human's way of answering to a question and then analyze the subtle differences in the human way of analyzing things and a machine's perspective. Given a context paragraph, the basic way of answering the questions that follow the paragraph is that we read the question, then understand what is being asked and then try to search for the answer from the paragraph with the help of matched words of questions and paragraph. Problem will not arise for human beings even in the case of nested sentences. But imagine a machine given a passage to learn and understand and then answer the questions. A naive answer would be to run the string matching algorithm over the entire passage for a question. Practically, this method would not work. The most general cases where the machine fails is when answers do not occur in the sentences directly or more than one sentence can provide the equal number of strings matched for a question. The machine also fails in the questions of drawing conclusion and summarizing the paragraphs. Moreover for the string matching to work we need that our machine remembers the question and then search for all sentences in the passage. We cannot expect our program to use large amount of memory for this.

MRC model gives solution to all these problems.The model working on the principles of Neural Network Attention, Bi-Directional Recurrent Neural Networks is the widely used approach. Since Neural Networks is assumed to be replica of human brain cells we can assume that the developed model can always learn from the training set. Utilizing GRU cells we can assume that the model has a small amount of memory.

II. Related Works

Earlier models utilized one hot encoding method to encode the words in the dataset. But it has many drawbacks. The length of the vector needed to represent a word is equal to the number of words in the whole dictionary which is in millions. This sparse representation of words consumes large memory and thereby increasing the complexity of the model and reducing the efficiency of the model.

To overcome the above drawbacks, few models came up with the concept of Word Embeddings. These are the dense representation of words with each word represented by a fixed length vector. Word Embeddings have been used widely in Natural Language Processing models to represent the text in mathematical form.

Attention mechanism has also been used widely in comprehension models. We have passage representation and question representation separately. But we need to look at them together. It is here where the attention mechanism comes to picture. Earlier models utilized the dot product attention where we perform dot product of the question and passage representations. More complex attention mechanisms are BiDAF

attention which states that attention flows in two directions - question to passage and passage to question.

Our model is different from the earlier models in utilizing the latest version of RNN cells - GRU cells. Gated Recurrent Unit cells are more efficient in storing the context of the passage in its long term memory and they have less number of gates (2- update and reset gates) when compared to LSTM cells (3- update, forget and output gates). GRUs are computationally cheaper. We also utilize the self matching attention mechanism to obtain the combined mathematical representation of question and passage.

Our model also utilizes Pointer networks to find out the boundary of the answer for a given question which were not used in earlier models on MRC.

III. Task Description

The MRC model task can be defined as follows: For a given paragraph P := [p_1,p_2....p_n], question Q := [q_1,q_2....q_m], the model predicts answers A := [a_1,a_2....a_p] for all questions, where each answer a_i consists of starting and ending positions of the predicted answer.

IV. MRC Structure

A. Word Embeddings

Machines can only understand in binary, i.e., only 0s and 1s. The data we feed to the machine for training and testing are pure text that cant be understood by a computer. A simple and efficient solution to this problem is to represent the words using word embeddings. It is a technique in Natural Language Processing that maps the words to fixed size dimensional vectors.

A traditional method of representing the words is one-hot vector method, with its target element being 1, others being 0. The main drawback of this method comes when we look for uniqueness of each word in the vocabulary. Since English dictionary consists of millions of words, each word representation would require million bits of memory, which is practically infeasible.

Word2Vec is an efficient solution to this problem which converts the words into its respective word embeddings.

B. Bi-Directional Recurrent Neural Networks

Recurrent Neural Network is a deep learning technique that is used for sequential data such as text, voice, stock prices, etc. This is the most powerful of all kinds of Neural Networks because it has internal memory, capable of remembering the important things about the input they received and thus, giving a precise prediction. In RNN, the information cycles through a loop and when a decision has to be made it considers the present input and the training which it has learned from previous inputs. In RNNs we have the problem of vanishing gradient, i.e., it fails to back propagate the error to the initial layers of the network.

To resolve this problem, we have Gated Recurrent Units (GRUs). These are invented in 2014. These have two gates namely, update gate and reset gate. These are the vectors which makes the decision of the information to be given as output. The best feature of GRUs is that they can be trained to keep

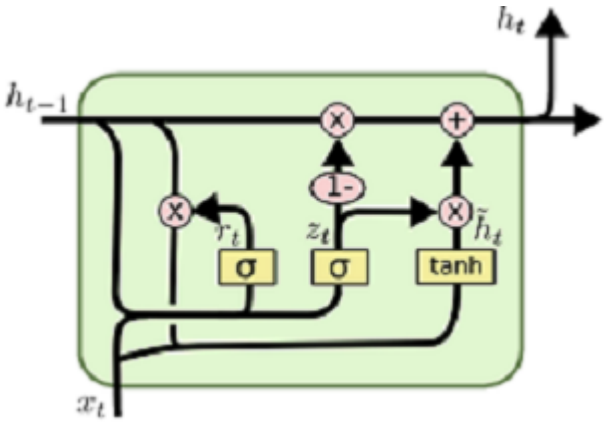

Fig. 1. GRU Working Unit

the information from long ago. If trained properly, the GRUs can have even complex applications. The structure of a GRU cell is shown in Fig 1.

The meaning of a word in a sentence depends on the words that precede and succeed it. So we need to train the model from both directions of the sentence. For this purpose we use Bi-Directional RNNs. The working equations of the GRU cell are shown below:

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

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

$$h^-_t = tanh(x_r U^h + (r_t * h_{t-1}) W^h) \tag{3}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * h^-_t \tag{4}$$

C. Attention

When a neural network is given a long sentence, due to its short memory handling capacity, some of the relevant information may be lost. To work on this problem, the seq2seq approach generates a vector. Whenever a new word is given as input, the network dedicates some of its part to search for the input token in the context paragraph. The context vector highlights the part of the sentence and sends only the relevant information. It focuses the networks attention. With this, the decoder network can receive relevant information from Bi RNN. The sentence length matters less because only a few words are considered relevant. The working equations of attention mechanism are shown below:

$$\alpha_{ts} = exp(score(h_t, h^-_t)) / \sum_{s=1}^{S} = exp(score(h_t, h^-_t)) \tag{5}$$

$$c_t = \sum_{S} \alpha_{ts} * h^-_t \tag{6}$$

$$a_t = f(c_t, h_t) = tanh(W_c[c_t; h_t]) \tag{7}$$

D. Pointer Networks

Pointer Networks are sequence to sequence models where the output is not a complete sentence but some discrete tokens corresponding to the input These are suitable for problems like sorting, word ordering, or computational linguistic problems. In all these problems the size of the dictionary is in proportion to the length of input string. Pointer Networks solve the problem of variable size output dictionaries using a mechanism of neural attention. It uses attention as a pointer to select a member of the input sequence (passage) as the output (answer).

V. Work Flow

We start with analysis of dataset and then convert the passage and questions to its word level and character level embeddings using NLP. The entire training of the model takes place in three readings.

In the first phase, we feed these embeddings to a bidirectional RNN.

In the second pass, the network trains itself with the context of the question using question matched attention.

In the third pass, the network finds the answer to the question and ignores the rest of the passage.

Finally, we use the concept of Pointer networks to find the starting and ending point of the answer to obtain the relevant answer as output.

The entire work flow can be visualized in Fig 1.

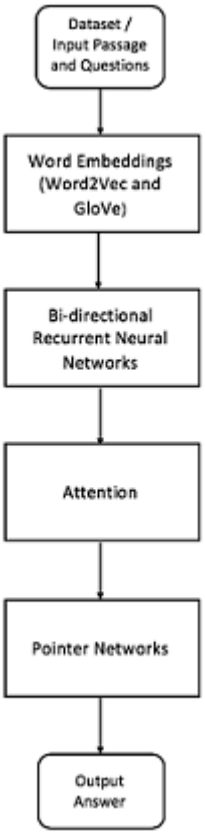

Fig. 2. Flow chart depicting the work flow to be carried out to develop MRC model

VI. Experiments

A. Dataset Analysis

Dataset analysis is an important parameter to derive the network parameters (number of cells in the network layer). SQuAD dataset consists of nearly 0.1M question-answers along with associated paragraphs.The SQuAD dataset has been selected as it is more effective and quantitative than previously used smaller datasets for training of this model. To get used to the structure of dataset, we run a basic statistical analysis of the SQuAD Dataset. The first inspection of dataset was to understand the question answer pattern and the paragraphs. The question and answer length were also analyzed.

B. Word Embeddings

We imported the passages, questions and corresponding answers from the SQuAD dataset and parsed them to sentence tokens and the sentence tokens to word tokens. We then imported pretrained GloVe Embeddings and extracted the embeddings of nearly 6B tokens with each token represented with its corresponding 100 dimensional vector. We obtained the similarity in meaning between the two words using the "Cosine Similarity" measure. Cosine similarity between the two words is the dot product of the vector representations of the corresponding words.

Cosine Similarity can be calculated with the help of following formula :

$$CosineSim(vec1, vec2) = (vec1.vec2)/(|vec1| * |vec2|) \quad (8)$$

The similarity between a word with itself is 1. We also found out the most similar word to a given word by computing the cosine similarity between the given word with all the other tokens of the dataset and obtaining the word with maximum similarity.

C. Training Of Neural Network

With the help of glove embeddings, the words in both passage and question were converted to its respective embeddings (vectors). The passage vectors were fed first into an Bi-Directional GRU layer and vector representation of the entire passage was obtained. To the same GRU layer, the question vectors were fed and the vector representation of the entire question was also obtained.

The two different representations of passage and question were combined to obtain the passage-to-question attention and question-to-passage attention.

These representations were passed to another Bi-Directional GRU layer to refine the memory content.

D. Attention

The starting and ending point of the answer were predicted using attention vectors and memory content with the help of pointer networks.

VII. Results

The data analysis on the training dataset for question type distribution is summarized in a pie chart as shown in Fig 2. The major number of questions are of what type and least number of questions are of how type.

The test dataset follows almost similar type of question distribution as the training set as shown in Fig.3.

The average length of each question type and its answer were found out and plotted in the form of a bar chart as shown in Fig 4. Seven types of questions were taken into consideration and rest of the questions were considered as other type. The WHY type of questions have much higher answer length when compared to the other type of questions.

The test set also follows a similar distribution as shown in Fig 5

The various statistical aspects (like word count, title count, average question and answer lengths, etc.) were recognized and were drawn out of the dataset and the results were shown in Table 1.

Cosine similarity between selected pairs of words were found out using Eq.1 and were tabulated as shown in Table 2.

Most similar word for a few selected words were found out and were tabulated as shown in Table 3.

Two metrics were used to evaluate the performance of the model - Exact Match and F1 score.

Fig. 3. Distribution of different types of questions in the training set

VIII. Conclusion

In this paper, we explored the problem of Machine Reading Comprehension and proposed a solution to represent the words in the SQuAD dataset in mathematical form using word embeddings. Firstly, we performed the basic statistical analysis on the SQuAD dataset. Working of word embedding representation of words were presented using the cosine similarity measure.

Fig. 4. Distribution of different types of questions in the test set

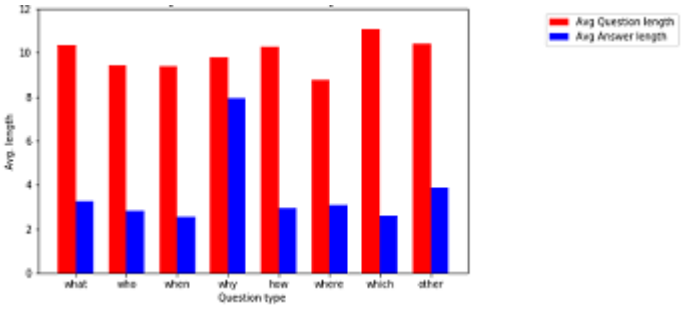

Fig. 5. Average length of different types pf questions and answers in training set

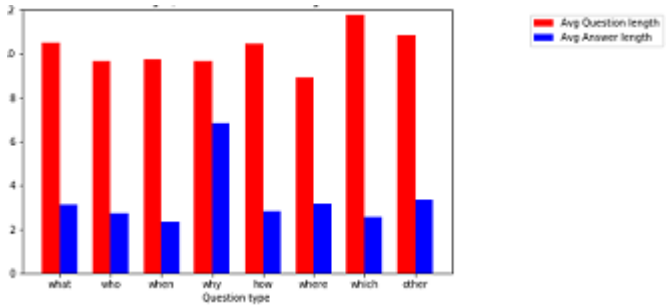

Fig. 6. Average length of different types pf questions and answers in test set

TABLE I
Exploratory Data Analysis Table

Sl.NO	*Statistical Aspect*	*Training Set*	*Test Set*
1	Title Count	442	48
2	Total Word Count	91,87,544	10,93,051
3	Total Question Count	87,599	10,570
4	Context Count	18,896	2,067
5	Average Question Length	10.46	10.82
6	Average Answer Length	3.26	3.15
7	Average Context Length	117	123
8	Average Question count per context	4.64	5.11

TABLE II
TABLE DEPICTING THE COSINE SIMILARITY BETWEEN WORD 1 AND WORD 2 AS PER THE CONTEXT OF PARAGRAPH

Sl.NO	***Word 1***	***Word 2***	***Cosine Similarity***
1	Cricket	Football	0.666
2	Cricket	Sport	0.485
3	Banana	Mango	0.705
4	Fruits	Sports	0.077
5	Football	Banana	0.142
6	Bat	Ball	0.639
7	College	School	0.885
8	Pen	Pencil	0.610

TABLE III
TABLE DEPICTING THE MOST SIMILAR WORD FOR A GIVEN INPUT AS PER THE CONTEXT OF PARAGRAPH

Sl.NO	***Word***	***Most Similar Word***
1	Cricket	Rugby
2	Sports	Sport
3	Apple	Microsoft
4	Mango	Papaya
5	Eraser	Pencil
6	Bat	Ball
7	College	School
8	Pen	Pencil

In the next phase of developing our model we fed the vectors to the bi-directional RNN i.e GRU system and trained the model. The concept to attention was also used to maintain a parallel viewing of context of paragraphs and questions simultaneously. This model has various applications in text processing problems like text summarization, speech-to-text translation, chat bots, etc.

IX. ACKNOWLEDGEMENT

We would like to express our sincere thanks and gratitude to Prof. Arun Kumar for his invaluable guidance and continuous support throughout the project.We would also like to extend our gratitude to Prof. Bidyut Kumar Patra for providing guidance and necessary requirements for the project.

REFERENCES

[1] Wissam Baalbaki, Dan Zylberglejd ; "Natural Language processing with Deep learning Reading Comprehension"
[2] Natural Language Computing group, Microsoft Research Asia; "R-NET: Machine Reading Comprehension with Self Matching Networks"
[3] Chao Wang, Hui Jiang; "Exploring Machine Reading Comprehension with Explicit Knowledge"
[4] FNU Budianto; "Reading Comprehension on the SQuAD Dataset".
[5] Yining Hong, JialuWang, Yuting Jia, Weinan Zhang, XinbingWang; "Academic Reader: An Interactive Question Answering System on Academic Literatures"
[6] Alvaro H. C. Correia, Jorge L. M. Silva, Thiago de C. Martins, Fabio G. Cozman; "A Fully Attention-Based Information Retriever"
[7] Suryani Atan; "Adaptive Reading Comprehension (ARC) System for Singapore Malay Language Students", 2018 International Conference on Machine Learning and Data Engineering (iCMLDE)
[8] Chen Zhang, Xuanyu Zhang, Hao Wang; "A Machine Reading Comprehension-based Approach for Featured Snippet Extraction", 2018 IEEE International Conference on Data Mining
[9] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. "Dropout: A simple way to prevent neural networks from overfitting". The Journal of Machine Learning Research, 15(1):19291958.