Natural Language Processing - 585

Fall 2017

Project Report

Abhay Doke, Ravi Agrawal

**ABSTRACT**

Machine Comprehension (MC) has always been a challenging task from a Natural Language Processing perspective. It can be defined as answering a query about a given passage of text. To be able to answer questions from a given text, the Reading Comprehension system needs to model complex interactions between question and the passage text. Various techniques and models are proposed to solve this problem of Machine Comprehension and in this project we study the famous Bi-Directional Attention Flow Model (BiDAF). We retrained the BiDAF model and were able to reproduce results of the original model. We present some interesting visualizations for these results. Also, this model is very large and it takes quite long time to train. To make the training faster, we made some changes in the sizes of various layers in the model by carefully looking at training dataset. We present our results and compare these results with the original Bi-Directional Attention Flow Model on the Stanford Question Answering Dataset (SQuAD).

1. **INTRODUCTION**

Machine Comprehension of text has been a core research topic in the Artificial Intelligence and Natural Language Processing community for many years. Recent advancement of computing resources and the widespread use of mobile devices has given birth to many interactive applications such as such as Google Now, Amazon Alexa, Apple Siri which are question-answering systems at the core. Question-answering systems have great economic importance when used as smart chat bots in automated customer service applications and this has boosted the research related to Machine Comprehension.

**What is Machine Comprehension?** Over the years, many people have proposed various definitions of Machine Comprehension. Initially it was defined by invoking the famous Turing Test which says “*a machine attains human level intelligence if its responses in a dialog with a human are indistinguishable from those of another human” (Turing, 1950).* This is a very broad definition from Machine Comprehension task’s perspective and over the period of time many people have refined its definition by restricting the scope of the Machine Comprehension Task. Most widely accepted definition of Machine Comprehension is given by Christopher J.C. Burges. He defines the task of Machine Comprehension as, *"A machine comprehends a passage of text if, for any question regarding that text that can be answered correctly by a majority of native speakers, that machine can provide a string which those speakers would agree both answers that question, and does not contain information irrelevant to that question.”*[<https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/mct.pdf>].

Core requirement to solve this problems is to be able to understand the context text, understand the query, study the correlation between context and the query, and get the answer correctly. To be able to answer questions from a given text, the Reading Comprehension system needs to model complex interactions between question and the passage text. Also it should have many complex skills such as coreference resolution, analogy detection, understanding spatiotemporal relations etc.

Many successful models have been proposed to tackle the Machine Comprehension task. Most of the successful models have incorporated the attention mechanism which enables the Machine Comprehension model to intelligently focus on certain part of the context paragraphs which is most relevant to answer the question. Bi-Directional Attention Flow (BiDAF) model is one the popular model for the Machine Comprehension task and we study this model in this project. We have also tried changing the architecture of BiDAF model to make the training faster and we present our results.

1. **RELATED WORK**

**2.1 Literature Review**

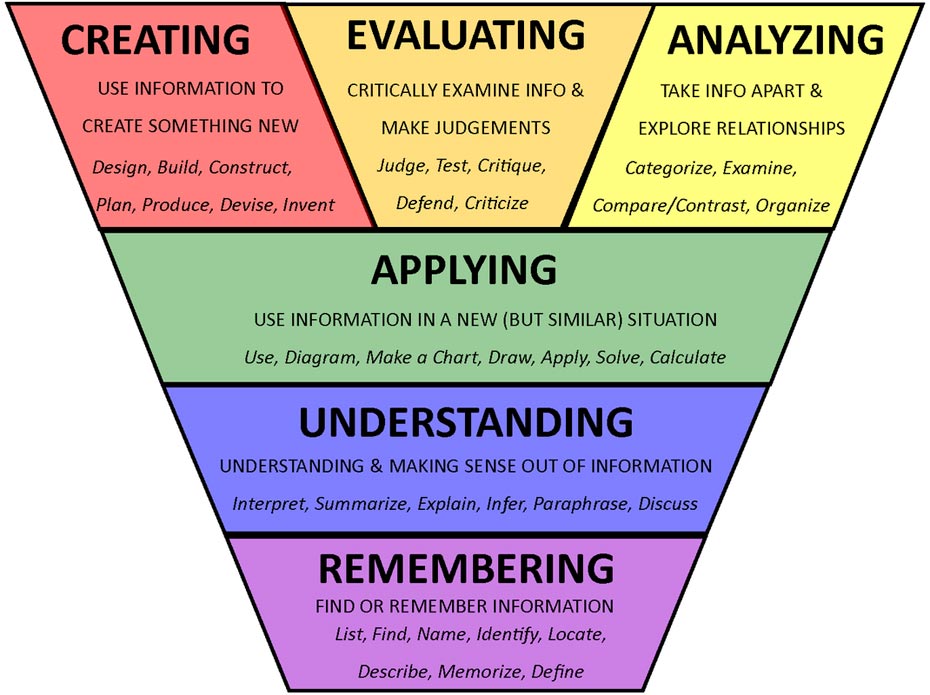
The SQuAD dataset is used widely to develop different deep learning model. From the the SQuAD website leadership board [2, 7, 8, 9] clearly demonstrates that almost all the top model utilizes some form of the Attention model. The Attention model was first introduced in the Neural Machine Translation [6]. This model utilizes to the whole original sentence states and not just the last state, to generate a translation word. The idea is then applied to the reading comprehension, allowing the model to select a subset of context paragraph and a subset of question that are most relevant. That way, the model uses the most relevant information to give a better answer [1].

Convolution Neural Network was also used for the task of machine comprehension [10]. This paper used pre-trained word vector for the sentence- level classification task. There CNN model with little hyperparameter tuning and static vector achieves excellent results on multiple benchmarks. Dynamic Coattention Networks have also tested for the task. This model consists document encoder, question encoder, coattention encoder and dynamic pointer decoder. The coattention encoder is the attention mechanism for this model. It encodes the interaction between the encoded question and the encoded document. The dynamic pointing decoder uses highway maxout a few times, using the information of the previous prediction to improve the next prediction. This iterative process allows the model to escape from a local maxima.

Another successful model for the Machine comprehension task uses Bi-Directional Attention Flow for Reading Comprehension (BiDAF) [5]. This is a multi-stage hierarchical process that represents the context at different levels of granularity and uses bi-directional attention flow mechanism to obtain a query aware context representation without early summarization. This model was able to achieve the state-of-the-art results in the stanford question Answering Dataset (SQuAD). This is the model we will be implementing for the current project.

**2.2 How good are the current Machine Comprehension models?**

To answer this question we need to look at the core of the Machine Comprehension task. Given a text, there can be various types of questions that one can ask. Difficulty levels of these questions can be categorized using the popular Bloom’s Taxonomy. Bloom’s taxonomy classifies the educational learning objectives into levels of complexity and specificity. From the Artificial Intelligence’s perspectives, we can use Bloom’s taxonomy to comment about how well the state of the art Machine Comprehension model is doing.



**Fig 1. Bloom’s Taxonomy**

Learning capacity of all state of the art models falls into the first level of the Bloom’s Taxonomy, where questions are factual. These models are basically locating words in the context paragraphs which answer the factual questions. There are some models which can summarize a given text as well.

Examples of questions in taxonomy:

* 1. **Remembering**
     1. Who spoke to...?
     2. When did … happen?
  2. **Understanding**
     1. Can you write in your own words...?
     2. What was the main idea...?
  3. **Applying**
     1. Can you apply the method used to some experience of your own...?
     2. From the information given, can you develop a set of instructions about...?
  4. **Analyzing**
     1. How was this similar to...?
     2. Can you distinguish between...?
  5. **Evaluating**
     1. Can you distinguish between...?
     2. How would you feel if...?
  6. **Creating**
     1. Can you create new and unusual uses for...?
     2. Can you write a new recipe for a tasty dish?

Consider an example passage of text from SQuAD dataset:

“*The Black Death is thought to have originated in the arid plains of Central Asia, where it then travelled along the Silk Road, reaching Crimea by 1343. From there, it was most likely carried by Oriental rat fleas living on the black rats that were regular passengers on merchant ships. Spreading throughout the Mediterranean and Europe, the Black Death is estimated to have killed 30–60% of Europe's total population. In total, the plague reduced the world population from an estimated 450 million down to 350–375 million in the 14th century. The world population as a whole did not recover to pre-plague levels until the 17th century. The plague recurred occasionally in Europe until the 19th century.*”

**Question:** “*When did the world's population finally recover from the black death?*”

**Possible Answers:** “*the 17th century, 17th century*”

**Question:** “*For how long did the plague stick around?”*

**Possible Answers:** *“until the 19th century, 19th century*”

These questions are factual, which means there is only one correct answer, which can be verified by referring to the context paragraph. So even though the current models have achieved promising results, they are only able cover the first level of understanding i.e. Knowledge as specified by the Bloom’s Taxonomy. These models won’t be able to answer questions from higher levels of Bloom’s Taxonomy.

1. **DATASET**

We are using SQuAD dataset for our experiments. SQuAD dataset can be found at “<https://rajpurkar.github.io/SQuAD-explorer/>”. Firstly, we took a deeper dive into the dataset and tried to explore and visualize the dataset. Dataset comes in json format. Size of the training data is 30MB and dev data is 5MB

Training Dataset:

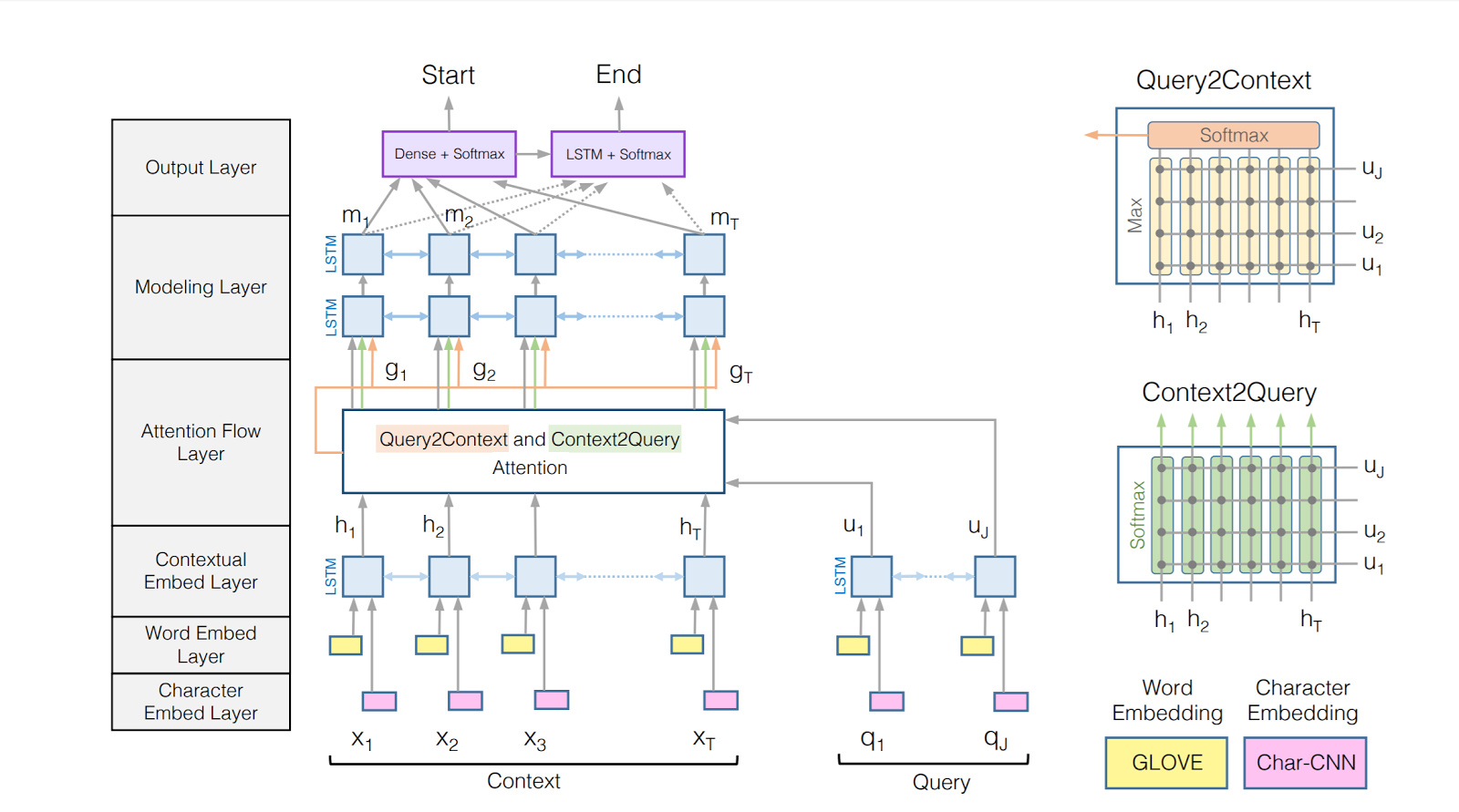
We explored the training data more deeply to gain more insights about the data. Training data is a JSON file and is divided into 442 topics. For every topic there are multiple context texts provided and for every context text there are multiple questions and answers are collected. Table 1 provides a subset of topics in the training dataset. It can be seen that there are lot of variety of topics in the training dataset. These topics cover texts about science, fiction, places, geography, literature etc. Because of this large coverage of topics, SQuAD dataset is treated as the state of the art dataset for the machine comprehension task.

For every topic there are number context paragraphs are provided and this number vary for every topic. Table 2 shows some of the topics with maximum and minimum number of topics. In the training dataset there are total 18896 context paragraphs. Lengths of these context paragraphs vary a lot and Figure 3 shows a distribution of number of words in all context paragraphs. Also we looked at the number of words in all questions and answers as shown in Figure 4 and Figure 5.

We conducted two experiments, 1. training original BiDAF model and 2. changing the layer sizes of original BiDAF model to make the training faster. For the second experiment dataset statistics helped us to decide layers’ sizes of Bidirectional Attention Flow Model. Since most of the context paragraphs have less than 300 words, size of the *context layer* (as shown in Figure 2) is set to 300 initially. We experimented with these numbers and tried to see if an increased size of the context layer helps in better performance. Similarly, for deciding the size of the *query layer* (as shown in Figure 4) is set to 30. In addition to the answer text, starting position of that answer in the context paragraph is also given. This information is used to train the answer localization network in the Bidirectional Attention Flow Model.

Dev Dataset:

Dev dataset is used as a test dataset which is different from hidden dataset on the leader board. In the dev data, there are total 2067 context paragraphs from 48 different topics. There 10570 questions with 34736 answers. Each question has one or more than 1 possible answers.



**Figure 2.** Bidirectional Attention Flow Model

1. **METHOD**

In this project experiment we used the BiDAF model for the training. We tried to reproduce the results of original BiDAF model. The Bidirectional Attention Flow Model for the reading comprehension is a hierarchical multi-stage process consisting six layers:

1. **Character Embedding Layer**

Character embedding layer is responsible for mapping each word to a high dimensional vector space. This model acquires the character-level embedding of each word using convolutional neural networks (CNN). Characters are then embedding into the vectors, which can be considered as 1D inputs to the CNN, and whose size is the input channel size of the CNN. The outputs of the CNN are max-pooled over the entire width to obtain a fixed-size vector for each word.

1. **Word Embedding Layer**

Word embedding layer also maps each word to a high dimensional vector space. In this project we are using pre trained word Vectors, GloVe (Pennington et al., 2014) to obtain the fixed word embedding of each word.

The concatenation of the character and word embedding vectors is passed to a two-layer Highway Network (Srivastava et al., 2015). The outputs of the Highway Network are two sequences of d - dimensional vectors, or more conveniently, two matrices: X ∈ Rd×T for the context and Q ∈ Rd×J for the query.



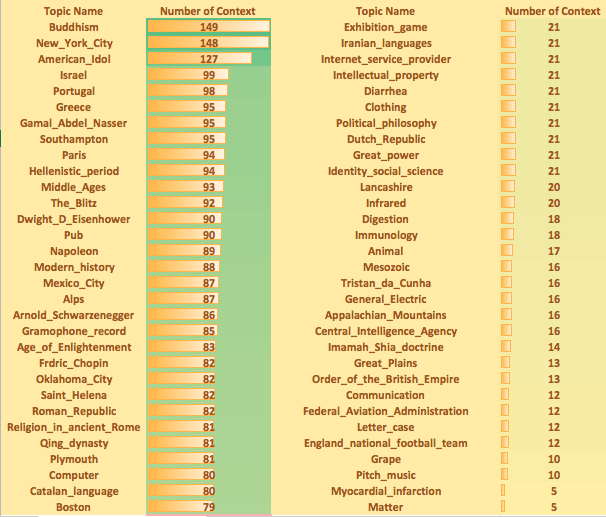
**Table 1.** Example topics from the dataset

1. **Contextual Embedding Layer**

Bidirectional Attention flow model also utilizes Long Short-Term Memory Network (LSTM) on top of the embeddings provided by the previous layers to model the temporal interactions between words. LSTM is placed in both directions, and concatenate the outputs of the two LSTMs. Hence we obtain H ∈ R2d×T from the context word vectors X, and U ∈ R 2d×J from query word vectors Q.

1. **Attention Flow Layer**

Attention flow layer is responsible for linking and fusing information from the context and the query words, the attention flow layer is not used to summarize the query and context into single feature vectors. Instead, the attention vector at each time step, along with the embeddings from previous layers, are allowed to flow through to the subsequent modeling layer. This reduces the information loss caused by early summarization.



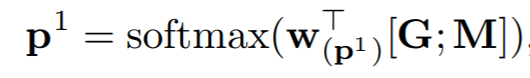
**Table 2.** Left column shows top 31 topics with maximum number of context paragraphs and right column shows topics with minimum number of context paragraphs.

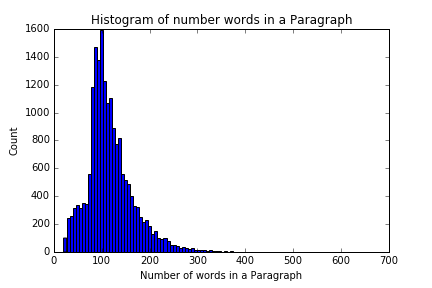
1. **Modeling Layer**

Modeling layer encodes the query-aware representations of context words. The output of the modeling layer captures the interaction among the context words conditioned on the query.

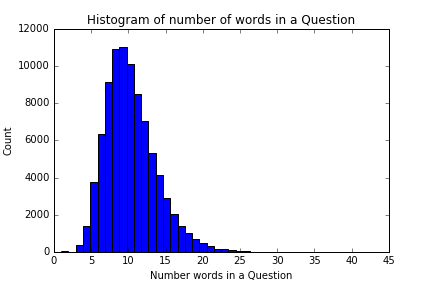
1. **Output Layer**

The QA task requires the model to find a sub-phrase of the paragraph to answer the query. The phrase is derived by predicting the start and the end indices of the phrase in the paragraph. We obtain the probability distribution of the start index over the entire paragraph by



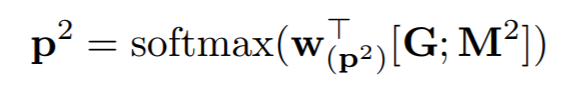


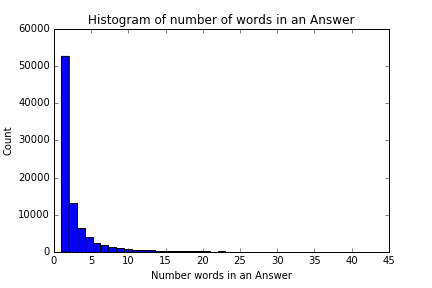
**Figure 3.** Histogram of number of words in a Context Paragraph



**Figure 4.** Histogram of number of words in a Question

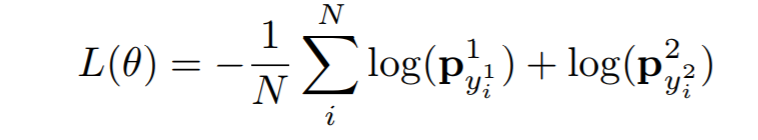
where w(p1) ∈ R 10d is a trainable weight vector. For the end index of the answer phrase, we pass M to another bidirectional LSTM layer and obtain M2 ∈ R 2d×T. Then we use M2 to obtain the probability distribution of the end index in a similar manner:





**Figure 5.** Histogram of number of words in an answer

**Training:** Training loss (to be minimized) is defined as the sum of the negative log probabilities of the true start and end indices by the predicted distributions, averaged over all examples:



Where θ is the set of all trainable weights in the model (the weights and biases of CNN filters and LSTM cells), N is the number of examples in the dataset, y1i and y2i are the true start and end indices of the i-th example, respectively, and pk indicates the k-th value of the vector p.

1. **EXPERIMENTS AND RESULTS**

**5.1 Training Original BiDAF model**

We took a tensorflow model for Bidirectional Attention Flow Model and tried to run it over the SQuAD dataset. Code implementation and details can be found at: <https://github.com/imraviagrawal/ReadingComprehension>

Performance of the Reading Comprehension models is measured in terms of F1 score and Exact Match (EM) score. F1 score is the harmonic mean of precision and recall (the higher the better). For each question, precision is calculated as the number of correct words divided by the number of words in the predicted answer. Recall is calculated as the number of correct words divided by the number of words in the ground truth answer. The F1 score is computed per question and then averaged across all questions. EM score (the higher the better) is the number of questions that are answered in exact same words as the ground truth divided by the total number of questions.

**Figure 6.** F1 score comparison of reproduced results with original model results

**Figure 7.** Exact Match (EM) score comparison of reproduced results with original model results

**Figure 8.**

**Figure 9.**

1. **FUTURE WORK**

We currently have not run the model on the full dataset, By the end of this month we would like to run the model at least once in the full dataset. After training on the full dataset we will start testing if results can be achieved as per the original paper implementation of Bidirectional Attentional flow model. Once we get the similar results as that of the Bidirectional Attentional Model, we will try to further optimize the network. It takes large amount to time to train the model. On an average it takes 3 hours for a single epoch and we are trying to train the model at least for 10 epochs.