Machine Reading Comprehension for the SQuAD Dataset using Deep Learning

Chung Fat Wong

cfw20@stanford.edu

Abstract

Since the release of the Stanford Question Answering Data, a lot of work has been done on Machine Reading Comprehension using end-to-end deep learning networks. In this assignment, I have taken the proposed baseline architecture and investigated how performance may be improved through adding more sophisticated network components as well as fine-tuning of the hyperparameters. Through the accumulation of these improvements, the final single model achieved a score of 75.7% FI and 65.3% EM on the dev set.

1. Introduction

Machine reading comprehension is an interesting task as it is both difficult and useful. It is difficult as the machine needs to be able understand complex interactions between the question and the context paragraph in addition to the many well-known difficulties associated with Natural Language Processing. It is useful as a well-trained system might be utilized in the work place, e.g. question and answering of regulatory documents or as a personal assistant, e.g. handling information storage and retrieval.

In the following sections, we will go through the related research that been done on this topic as significant progress has been made in recent years, especially after the release of the SQuAD dataset [1]. We will describe the approach that we took to get to the final model, including the intuition behind some of the network components. Then we will go through the experiment set up of this project, including a detailed analysis of the results. And finally, future work that could be done to improve the performance of the model.

2. Related Work

There has been a lot of research work done on making better machine reading comprehension systems for SQuAD as can be seen by the very active leaderboard. One of the early successful models was the Bidirectional attention flow (BiDAF) [2] network which introduced an attention flow layer that allows both the context to attend to the question and vice versa. This attention layer is still regularly used in recent papers. The BiDAF network also sets out an architecture which consists of the following layers: (1) Embedding Layer, (2) Contextual Layer, (3) Attention Layer, (4) Modelling Layer and (5) Output Layer. We will also follow this general framework.

Another high-performing model is the Dynamic Coattention Network [3] which also has a layer allowing two-way attention between the context and the question. The Coattention layer also computes second-level attention which means attending over representations that attention outputs. In this assignment, we implemented both the BiDAF and Coattention layers and analyzed their effect on the performance of the overall network. In a slightly different direction, the R-Net [4] contains a self-attention layer which allows a sequence of representations to attend to all other elements in the sequence.

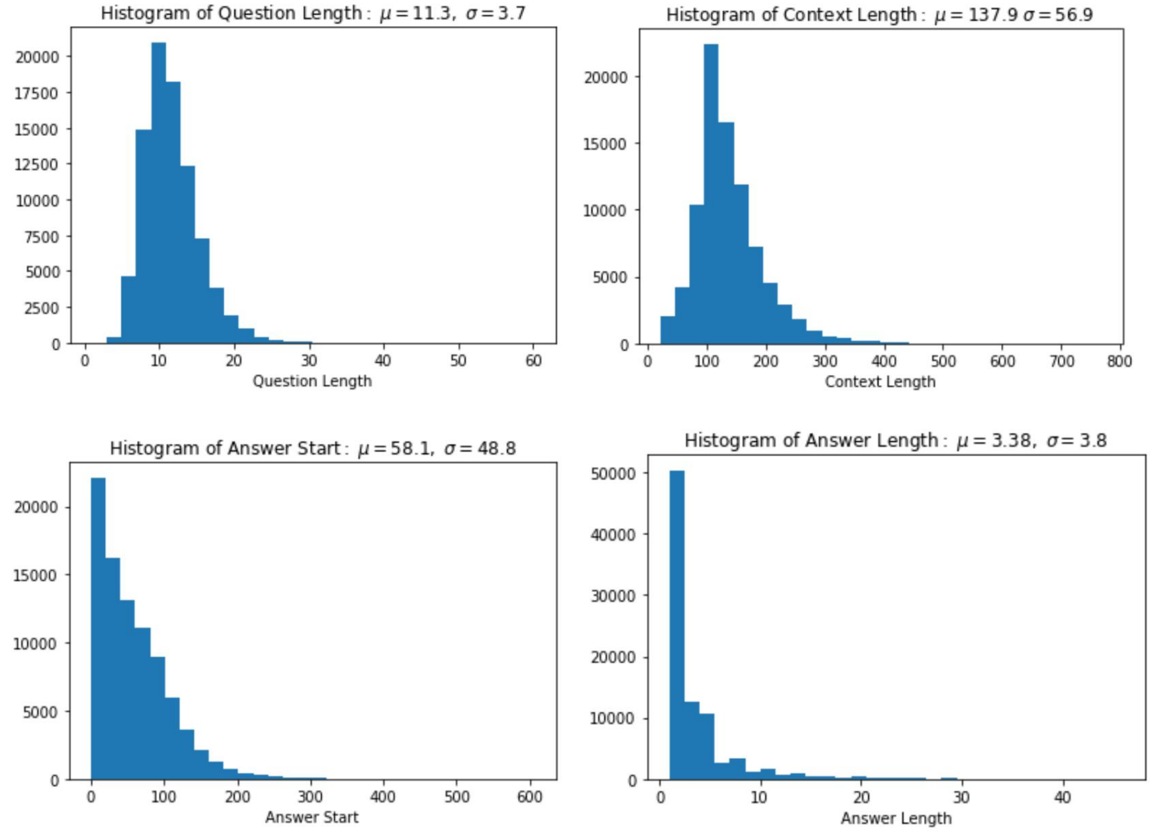
More recent architectures such as the one introduced in Simple and Effective Multi-Paragraph Reading Comprehension [5] combine various types of attention layers in order to allow more sophisticated representations of the context locations. In this particular case, the BiDAF layer is connected to self-attention layer. This is the implementation that I have followed in this assignment.

In addition to the above, we also used ideas from other research papers to try to improve the performance of specific parts of the network. For example, we used an idea from the DrQA [6] to achieve smarter span selection at test time. And the FusionNet [7] model provided information about how to combine different types of attention layers together as well as experimental data on the advantages and disadvantages of different attention functions, e.g. additive and multiplicative.

3. Dataset

The main dataset for this assignment is the Stanford Question and Answering Dataset (SQuAD). It is a reading comprehension dataset consisting of questions posed by crowdworkers on a set of Wikipedia articles. The answer to every question is a segment of text from the corresponding reading passage. The training set consists of around 86,000 question-answer pairs while the development set contains around 10,000. The test set is hidden.

It is informative to visualize various aspects of the training dataset as it gives a quick check on the ranges within which the hyperparameters should be searched.



For example, the histogram for the context lengths clearly show that the majority of these are shorter than 300 words (around 98%). Hence it is possible to set the maximum input length of the context for the input layer to around 300 without losing much in the data but gaining a lot in terms of memory efficiency. The maximum input length for questions could be set in the same way to around 20.

The histogram for the answer lengths show that the vast majority of answers are very short, e.g. 97% are below 15 words. This information helps with determining the answer span at test time as we could limit the search for answer spans to lengths between 0 and 15 to prevent the model from choosing unnecessarily long answer spans. This was implemented using the methodology suggested in DrQA and improved the dev test F1 score by around 2%. As the histogram for the location of the answer start also shows a strong bias towards the beginning of the paragraph, I also investigated whether this could be taken into account at test time, however the effect turned out to be much smaller than the one for the answer length.

Finally, looking at the dataset also gave a good argument that a specific set of the word embeddings should be trainable. The below table show the 20 most common words in question set vs the most common words in the context set.

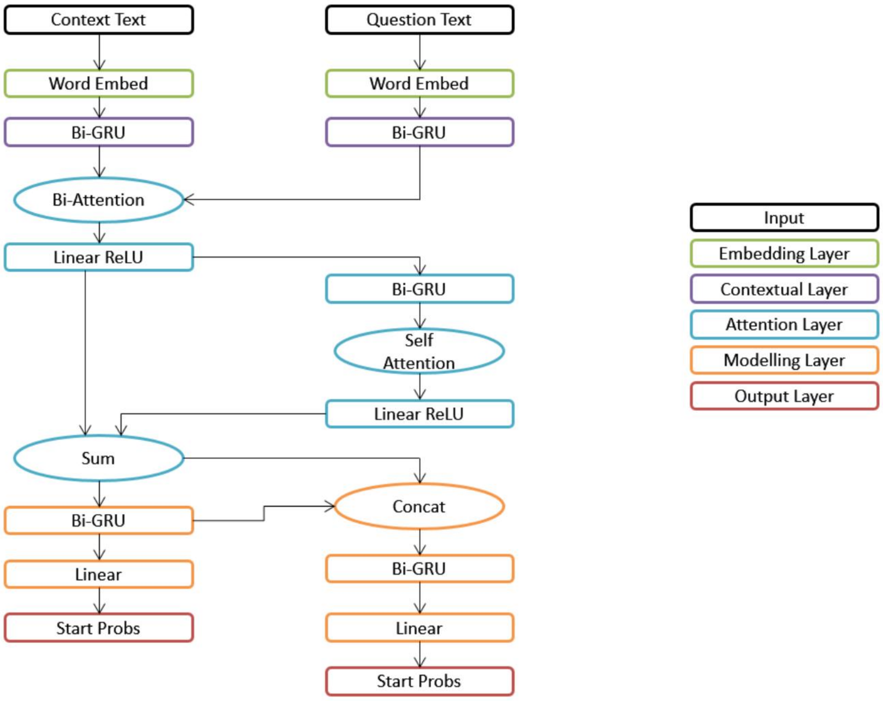
Common Question Word	Frequency
?	85,454
the	63,796
what	50,294
of	34,039
in	25,965
to	18,674
was	17,120
is	16,365
did	15,658
a	11,096
how	9,256
who	9,167
's	8,241
many	5,378

Common Context Word	Frequency
the	810,302
,	654,625
.	435,376
of	407,204
and	331,235
in	289,537
to	225,877
a	189,314
"	112,503
as	101,158
)	92,518
(	92,436
is	91,511
's	65,087

It’s clear that many of the common question words are not as common in the context set and given that the GloVe [8] embeddings are trained on the same data as the context set, i.e. Wikipedia articles, it seemed beneficial that some of the common question words should be trained in the question-answer environment. After some experimentation, 9 frequent question words were made trainable in the final model, including ['what', 'did', 'many', 'who', 'when', 'how', 'are', 'which', '?']. This improved the dev test F1 score by around 0.5%.

4. Model

The architecture for the final model is based on approach shown in Simple and Effective Multi-Paragraph Reading Comprehension. The organisation of the layers is shown in the diagram below. The following subsections will go through each layer in more detail.



Embedding Layer:

Embedding is done using pre-trained word vectors. Specifically, we chose to use GloVe.6B with 100 dimensions as experiment using higher dimensions did not improve performance. As described above, we made a specific set of frequently seen question words trainable. The vector representations for out-of-vocabulary words ('UNK') and padding ('PAD') are also made trainable.

Contextual Layer:

A shared bi-directional GRU is used to map the question and context embeddings to contextual aware embeddings

Attention Layer:

The attention layer combines both a BiDAF layer and a Self-Attention layer. Specifically, the output from the BiDAF is passed through a linear layer with RELU activations and then is passed through a bi-directional GRU before going into the Self-Attention layer. The output from the Self-Attention layer goes through another linear layer with RELU activations and are then summed with the original outputs of the BiDAF. The intuition behind this combination is that the BiDAF lets the question words attend to the context as well as letting the context words attend to the question words. This results in a rich representation of the context locations. Since the context can be long and distant parts of the text may rely on each other to fully understand the content, the Self-Attention layer then lets the enriched context attend to itself.

BiDAF Layer:

Let $\mathbf{h}_\mathbf{i}$ be the vector for context word *i*, $\mathbf{q}_\mathbf{j}$ be the vector for question word *j*, and n_q and n_c be the lengths of the question and context.

$$a_{ij} = w_1 \cdot h_i + w_2 \cdot q_j + w_3 \cdot (h_i \odot q_j)$$

where $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ are learned vectors.

The context –to-query vector $\mathbf{c}_\mathrm{i}$:

$$p_{ij} = softmax(a_{i,:})$$

$$c_i = \sum_{j=1}^{n_q} q_j p_{ij}$$

The query-to-context vector **q_c**:

$$m_i = \max_{1 \le j \le n_q} a_{ij}$$
$$p_i = softmax(m)$$
$$q_c = \sum_{i=1}^{n_c} h_i p_i$$

The final output is:

$$[h_i; c_i; h_i \odot c_i; q_c \odot c_i]$$

Self-Attention Layer:

Let $\mathbf{x_1}$, ..., $\mathbf{x_n}$ be a sequence of representations corresponding to context locations:

$$e_j^i = f(Ux_i)^T Df(Ux_j)$$
$$t^i = softmax(e^i)$$
$$b^i = \sum_{j=1}^{n_c} t_j^i x_i$$

Here *f(x)* = max(0, *x*) i.e. the ReLU function and D is a diagonal matrix. We use this symmetric form with nonlinearity to compute the attention function as the original additive attention is very memory intensive. Also it was shown in the application of FusionNet that this form of attention function performance better than the additive form. The resulting vectors are concatenated with the original vectors and passed through a linear layer with ReLU activations. Finally, those outputs are summed with the original outputs from the BiDAF layer.

Modeling Layer:

A last bi-directional GRU is applied, followed by a linear layer that computes the answer start logits for each word in the context. The hidden states of this GRU are concatenated with the inputs and fed into a second bi-directional GRU and linear layer to find the answer end logits. Both the answer start and end logits are softmaxed to produce probabilities. During training, we optimize the negative log-likelihood of selecting the correct start and end tokens.

Prediction:

Instead of taking separate argmaxes over p^{start} and p^{end} to get the predicted span (which could end up with the end location occurring before the start location in some cases), we implement the methodology in the DrQA paper. In this case, the aim is to find the start and end context location pair (*i, j*) such that $p^{start}(i)p^{end}(j)$ is maximized but subject to the constraint that $i \le j \le i + ans_{len}$

5. Experiments

Implementation:

For the implementation of the model, we built upon the Tensorflow code which was provided as the baseline for the assignment. As mentioned above, for the word embeddings, we use the pre-trained GloVe.6B vectors. The embeddings for ['PAD', 'UNK'] and ['what', 'did', 'many', 'who', 'when', 'how', 'are', 'which', '?'] are made trainable. The model is trained with the AdamOptimizer with a learning rate of 0.001. The hidden size for all the bi-directional GRU's and linear layers with ReLU activation is set at 200 and they all share a drop out rate of 20%. Batch size is set at 100. In the final version of the model, a batch of size 100 takes around 4.6 to 4.8 seconds to train on an Azure NV6 machine (vs 1.1 to 1.3 seconds for the baseline model).

Experiments:

In order to get the final state of the model, a large number of iterations of network components as well as the hyperparameters were performed. The largest contributors to the performance improvements over baseline were the bi-directional GRU's in the modelling layer and the substitution of BiDAF into the basic attention layer. Together these account for around 25% increase in the F1 score over the baseline model. The addition of a smarter span selection procedure added around 2-4% as did the training of the small number of word embeddings. The addition of the self-attention layer added around 1%. Other small improvements added <1% .

Results:

The SQuAD task is mainly evaluated using the F1 and EM scores which are clearly quantitative. However it is also important to look at the performance of the model qualitatively, e.g. by looking at example results. We will first look at the quantitative measures.

The F1 score is the less strict of the two quantitative metrics as it is the harmonic mean of precision and recall, i.e. it take into account the amount of overlap between the predicted answer and ground truth. On the other hand. The EM score would be 0 for any example where the predicted answer is not exactly the same as the ground truth. Our final model produced 75.70% F1 and 65.29% EM on the dev set. The below table compares these performance numbers to the leaderboard figures of the related works.

MODEL	F1 (%)	EM (%)
R-NET (SINGLE)	84.265	76.461
FUSIONNET (SINGLE)	83.900	75.968
BIDAF (SINGLE)	77.323	67.974
OUR MODEL (SINGLE, DEV SET)	75.700	65.289

It is also useful to examine how the model performs for different types of questions.

FIRST WORD IN QUESTION	F1 (%)	EM (%)
WHEN	82.95	76.56
WHO	76.41	70.35
HOW	74.44	65.21
WHAT	73.08	63.05
WHERE	72.83	62.11
WHY	59.84	37.86

There's a clear difference in performance of the model on questions starting with 'WHEN' vs questions starting with 'WHY'. There are two likely explanations for this: (1) questions starting with 'WHY' requires more reasoning and comprehension of the text which is supported by the fact that 'WHY' answer are average the longest (around 7 words vs 2-3 for other types) and (2) the number of 'WHY' in the dataset is small compared to the others, e.g. accounting for only 5% of the dev set, hence the model would not be as well trained for these.

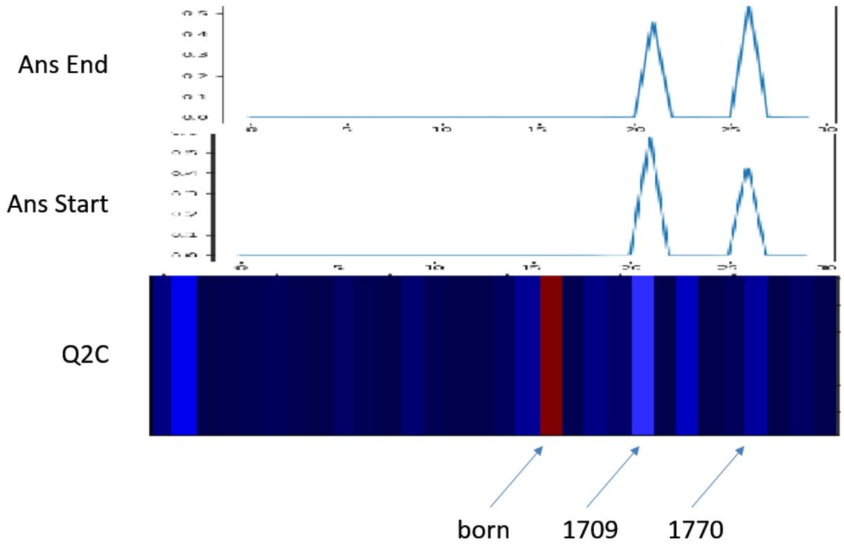
On the qualitatively side, it is useful to visualize the attention weights, the answer start and end probability distribution for both questions where the model answered correctly as well as the ones where it answered incorrectly. For the purposes of the visualisation, we will only show the BiDAF Q2C weights even though our actual attention layer is more complex as these weights tend to be more intuitive.

For example, the model scored 28.6% F1 on the following question and context set:

- Context: Charles Avison, the leading British composer of concertos in the 18th century, was born in Newcastle upon Tyne in 1709 and died there in 1770
- Question: What year did Charles Avison die in Newcastle?

- Predicted Answer: 1709 and died there in 1770
- True Answer: 1770

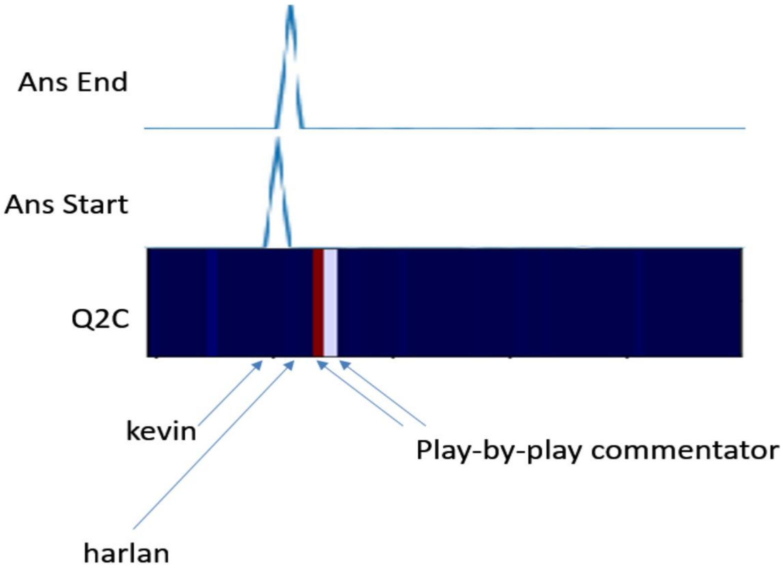
The model got this question incorrect as it included the birth year in the answer even though only the year of death was asked. The visualization shows that this might have happened because the attention was focused on the word 'born' in the context paragraph which probably caused both the answer start and end probabilities to have a significant weight on the year '1709' as well as '1770'. However, more work needs to be done to understand why the attention fell so heavily on 'born' in the first place.



On a question where the model got the exact match:

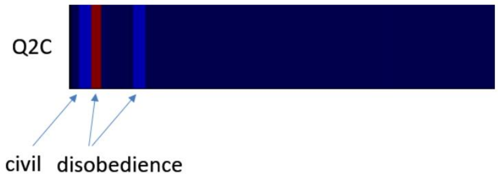
- Context: Westwood One will carry the game throughout North America, with Kevin Harlan as play-by-play commentator
- Question: Who is the play-by-play announcer for the game?
- Predicted Answer: Kevin Harlan
- True Answer: Kevin Harlan

In this case, it's clear that the attention did the correct job as it emphasised the words 'play-by-play' and 'commentator' even though the word 'announcer' was used in the question. And the answer start and end probabilities also gave the correct locations the highest weightings.



- Context: Non-revolutionary civil disobedience is a simple disobedience of laws on the grounds that they are judged "wrong" by an individual conscience, or as part of an effort to render certain laws ineffective, to cause their repeal, or to exert pressure to get one's political wishes on some other issue. Revolutionary civil disobedience is more of an active attempt to overthrow a government (or to change cultural traditions, social customs, religious beliefs, etc...revolution doesn't have to be political, i.e. "cultural revolution", it simply implies sweeping and widespread change to a section of the social fabric).
- Question: What type of civil disobedience is larger scale?
- Predicted Answer: non-revolutionary
- True Answer: revolutionary civil disobedience

The model got a 0 F1 score on this question. The attention weights show that there's focus on the words "civil" and "disobedience" at the start of the context paragraph which is actually sensible. However, the question requires a much deeper understanding of the meaning of both the context and the question, e.g. "larger scales" means to overthrow a government. Hence, for this type of question, it would be very difficult for machine question and answering systems currently.



6. Conclusion:

Building a machine reading comprehension system is an extremely difficult task even though we had the benefit of the significant research that has been done recently. Future experience with building end-to-end deep learning models will be useful in helping us to optimize the networks for peak performance. Fortunately, there are many avenues for future work:

- Implementing a character-level CNN to help with out-of-vocabulary words or use the larger variations of the GloVe embeddings
- Add POS and other features to the word embeddings
- Use ensemble of models
- Investigate new types of models e.g. 'Attention is all you need'

7. References:

[1] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text.

[2] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension

[3] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering.

[4] R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS

[5] Christopher Clark, Matt Gardner. Simple and Effective Multi-Paragraph Reading Comprehension

[6] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions.

[7] Hsin-Yuan Huang, Chenguang Zhu, Yelong Shen, Weizhu Chen. FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension

[8] Jeffrey Pennington, Richard Socher, Christopher D. Manning. GloVe: Global Vectors for Word Representation

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need