Question Answering on the SQuAD Dataset

Hyun Sik Kim
Department of Electrical Engineering
Stanford University
Stanford, CA 94305
hsik@stanford.edu
CodaLab: hsik

Abstract

Question answering is one of active areas in Artificial Intelligence (AI) research. It is a framework which concerns on how to build models that can extract an answer for a certain query on a context of paragraph. To tackle this challenge with the SQaUD dataset [1], I built deep recurrent neural networks combined with bidirectional attention flow and self-attention method. The single model achieved EM 66.11%, and F1 76. 302% on the dev set, and EM 67.481%, and F1 77.176% on the test set. The deep recurrent neural networks with attention mechanism showed very promising result on the question answering tasks on the dataset.

1 Introduction

Ever increasing amount of data triggered automatic machine comprehension models to be an active research topic. Because of its practical attributes, many eminent researchers have been investigated various methods for building powerful automatic question-answering models. Among many available approaches, recently, combining deep network structures and recurrent neural network frame works have shown promising results in real-world machine comprehension tasks. Especially, end-to-end deep recurrent networks can capture the information a query looking for from a passage, and can give reliable inference. However, these deep neural networks require huge amount of qualified training data, and the lack of training data made the task be challenging. To mitigate this problem, several datasets were publically distributed, and among them, the Stanford Question Answering Dataset (SQaUD) has been widely used for building question answering models because of its large number of question-answer pairs on Wikipedia articles. Recently, the deep neural networks trained on this dataset outperformed the human performance on comprehension.

Inspired by the successful results above, I aim to develop a novel question answering model trained on the SQuAD. This project was focused on tacking the question answering task with deep neural networks, especially, deep recurrent neural networks, and my implementation and empirical results of different deep recurrent neural networks applied to the SQaUD. I also compare the characteristics of network structures with different attention mechanisms, and different optimization objectives.

2 Related Work

The SQuAD has been actively used for developing a novel automatic question-answering model by lots of researchers. With sufficiently large amount of data, it has leaded the prosper of deep recurrent neural networks on this reading comprehension task.

Combining with the deep recurrent neural networks, the coattention mechanism [11] has been widely used, and contributed to the development of human-level machine comprehension models more recently. Comparing with the traditional attention mechanisms, the coattention is obtained by simultaneously considering all pairs of encoded information from both context words and question words, and this complex alignment allows a model to capture complex relationship between a query and a context paragraph.

Self-attention is an attention mechanism obtained by simultaneous alignment of input context itself. It has been widely used in variety of tasks, and gaining popularity especially in reading comprehension tasks. Combining with the coattention, the self-attention mechanism achieved state-of-the-art performance in question-answering tasks.

As the neural network structure gets deeper, the Highway network [8] has been utilized because of its effectiveness in constructing deep networks. The information flow within the graph becomes crucial for training deep neural networks, and the highway network can lessen the problem with deep networks by its gated network structure on information flow while preserving the dimension of the input.

The Bidirectional Attention Flow (BiDAF) [3] model is considered one of promising network architectures for deep neural machine comprehension models. The BiDAF utilizes the context-to-query attention, and query-to-context attention on encodings of a context and a query. The BiDAF allows hierarchical process of building a representation of a context with these bidirectional attention flow mechanisms. The BiDAF achieved high score on the SQuAD machine comprehension challenge.

This project is aimed to investigating different attention mechanisms, and deep neural network structures on the Stanford Question Answering Dataset, and evaluate the corresponding results.

3 Approach

In the following section, I describe how the model is formulated, and how the deep network works. The primary objective of the model is to capture the relationship between a query and a passage, and correctly localize a corresponding answer from them.

The network architecture of the best single model I found is illustrated in Figure 1. The network is a modified version of the BiDAF, and it accepts two input pairs, context embedding, and question embedding. Each embedding is composed of word-level embedding and character-level embedding. Word-level embedding can be obtained by pre-trained GloVe 6B 300-dimensional fixed word vectors. Character-level embedding is added to handle unknown words in the dictionary, and it is the output of a 1-d convolutional layer and max pooling layer. The output dimension of this character embedding layer is 100.

Now, the embedding vectors are expected to capture the semantics of each word. To have representations for both entire context and question, these two inputs are passed through a bidirectional Long Short Term Memory (BiLSTM). This bidirectional recurrent neural network allows the model to capture the meaning of entire context, and query separately. Since each embedding vectors for a query and a context are generated with the same source or method, I used the same BiLSTM layer for both query encoding and context encoding.

On these encodings, to capture the relationship between a query and a context of paragraph, I apply query-to-context attention, and context-to-query attention. The attention mechanism is basically the same for a query and a context, and I will present the context-to-query attention as an example:

$$\alpha_{ij} = \frac{e_{ij}}{\sum_{k=1}^{q_{length}} \exp\left(e_{ik}\right)}$$

$$e_{ij} = {C_{enc}}_i^T W {Q_{enc}}_j$$

$$CQ\ attention_i = \sum_{j=1}^{q_{length}} \alpha_{ij}\, {Q_{enc}}_j$$

Each encoded representation and its corresponding attention output are concatenated, e.g., [C; *CQ attention*; C⊙ *CQ attention*], and again passed through the next BiLSTM layer to have combined representations. The operation of ⊙ is elementwise multiplication, and it is inserted to increase the expressiveness of the model.

Since the outputs of the second BiLSTM are already processed by several nonlinear operations, it might lose its original information, so, to prevent information loss, I introduce the self-attention mechanism at this stage. The self-attention mechanism is identical to the previous attention method, but attention mechanism works on the encoded context encoding only:

$$\beta_{ij} = \frac{e_{ij}}{\sum_{k=1}^{c_{length}} \exp\left(e_{ik}\right)}$$

$$e_{ij} = {C_{enc}}_i^T W {C_{enc}}_j$$

$$Self\,CC\ attention_i = \sum_{j=1}^{c_{length}} \beta_{ij}\, {C_{enc}}_j$$

The final encoded vector will be: $[C_{enc}; C_{enc}Q_{enc}attention; C_{enc} \odot\ Self\,C_{enc}C_{enc}\ attention]$. This complex 3-d tensor will pass through the Highway network, and the same BiLSTM, and finally fully-connected layers build final representation. The series of complex layers are added to increase the expressiveness of the model, and these operations are designed to make the model capture the hierarchical context of both a query and a passage. The network is trained by minimizing the Softmax losses on the prediction of start and end tokens of a passage for a certain query.

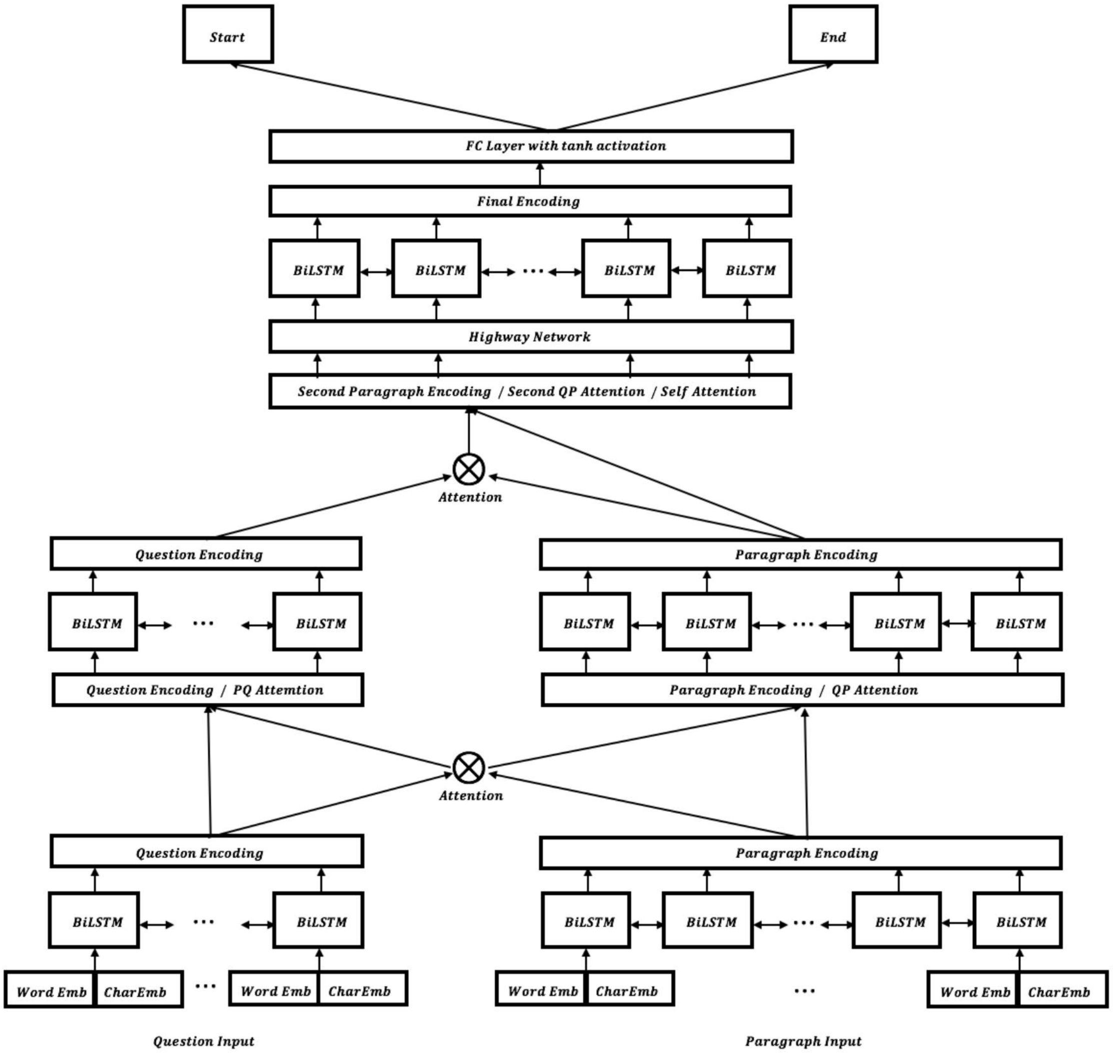

Figure 1: BiDAF model architecture

4 Experiments

In this project, I used the Stanford Question Answering Dataset (SQaUD) [1], and it is a large-scale question-answering dataset of over 100,000 question-answer pairs from over 500 Wikipedia articles. Table 1 below summarizes the dataset.

Table 1: Statistics of training data

SQuAD Dataset [1]	Training set (80%)	Validation set (10%)
# Question-Answer pairs	86,318	10,391
Minimum/Maximum length of contexts	22/766	24/700
Minimum/Maximum length of questions	1/60	3/34
Minimum/Maximum length of answers	1/46	1/37

Figure 2 illustrates the histograms of the training data. For efficient training, I set the maximum length of a context to be 300, and the maximum length of a question to be 30 to save training time. Also, I set the maximum length of an answer to be 15.

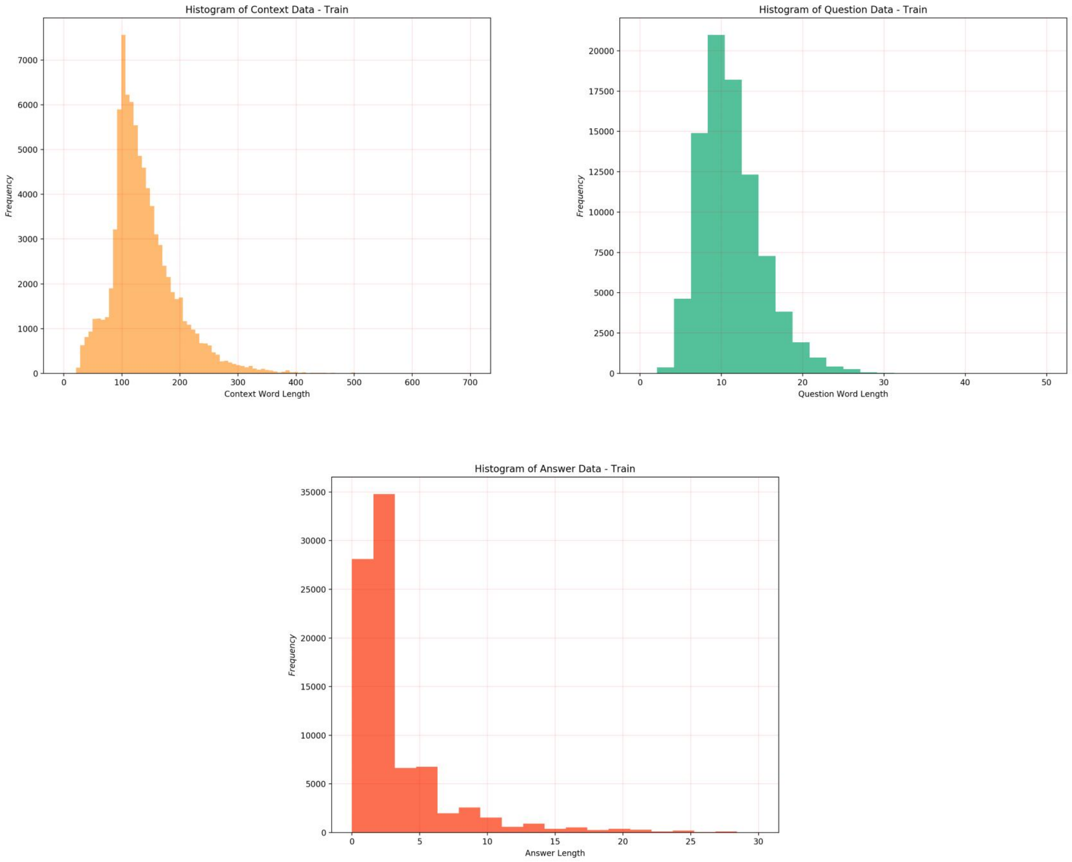

Figure 2: Histogram of training context, question and answer data

4.1 Training Procedure

I used the Adam optimizer with the initial learning rate of 0.002, and decaying it with the decaying factor of 0.95 every 500-iteration. The size of each hidden layer is 150 with the dropout rate of 0.2, and also I set the batch size as 150 to have faster convergence of training process. I used the pre-trained GloVe 6B 300d embedding, and it is fixed during the training. Also, I set the dimension of the character embedding as 20, and it will be processed with 100 width-5 kernels at the 1-d convolutional layer. I trained models on Nvidia Titan XP GPU, and used TensorFlow version 1.5. It took about 25 minutes on each epoch, and the best model is obtained around 25~30 epochs.

The evaluation metrics of this task are the F1 score, and the Exact Match (EM) score. The F1 score is calculated with the harmonic mean of precision and recall, and it allows some flexibility on measuring the performance of our model. The other metric, the EM score, is a fairly strict metric for the performance measuring, and it is calculated by checking whether the prediction matches exactly with the ground truth or not.

4.2 Summary of results

Table 2 summarizes the performance of models I developed. With appropriate attention mechanisms, the BiDAF structure itself showed very promising results. To handle with unknown words in a dictionary, my first trial was using a larger word embedding, the GloVe 840B 300d, but because of the out of memory issue, I changed my direction. Instead, I introduced the 1-d convolutional layer with a max-pooling layer [10], and this slight modification improved the dev set F1 and EM scores. When I made the pre-trained word embedding be trainable, the actual performance of the model is degraded, so I set the character embedding is trainable while used fixed the word embedding.

My single model of a modified BiDAF achieved the F1 score of 76.302%, and the EM score of 65.97% on the dev set, and **F1 77.176%** and **EM 67.481%** on the test set. And the ensemble of the 5 BiDAF models achieved 76.345%, and 66.11% respectively on the dev set. Unfortunately, I was not able to get additional performance improvement with the ensemble model, and this can be explained by the fact that the five models are already converged to similar local optimum points, and too short model saving period. If I increase the number of saved networks for the ensemble method, and use the bigger word embedding, e.g., the GloVe 840B 300d, I might be able to develop a model with the performance on par with the original implement of the original BiDAF model.

Table 2: The dev set results of BiDAF models

Own Model (Dev Set)	**F1 (%)**	**EM (%)**
BiDAF-LSTM-single	73.257	63.188
BiDAF-LSTM-Char_EMB-single	75.671	65.44
BiDAF-Highway-LSTM-single	76.302	65.97
BiDAF- Highway -LSTM-single	76.345 ↑	66.11 ↑

Table 3: Query type and performance of a BiDAF model

Query Type	**F1 (%)**	**EM (%)**	**# Questions**	**Avg. Prediction Length**	**Avg. Answer Length**
When	82.417	71.535	808	2.36	2.44
Who	74.045	64.988	1,231	2.76	2.83
How	70.505	51.820	1,154	2.62	2.97
What	68.940	53.653	5,763	3.30	3,26
Where	66.349	50.104	483	3.13	3.34
Why	56.529	24.667	150	7.45	6.75

4.3 Analysis on query type, answer length, and dataset statistics

Table 3 summarizes the performance of my implementation of the BiDAF on different query types, and the result is obtained from 10,000 question-answer pairs from the training dev set. From it, we can conclude that the model works well on short answer questions, while it does not perform well on "why" queries, and this might be explained by the fact that the length of the answer is relatively long, and requires high-level reasoning for answering the question. Even though this can be attributed to limited function of the attention mechanisms, it can be also caused by the unbalance in training data. Table 4 summarizes the histogram of the target answer in training dev set, and their length, and the corresponding F1 and EM scores. From Table 4, we can observe that almost 90% of the target answers are relatively short answers, and the number of longer answers takes very small amount in the total amount of the dataset. Combining with the data statistics and worse performance on queries with long answers, we can conclude that for better generalization and more powerful model, we need more question-answering pairs with the length of answers longer than 5.

Table 4: Answer length and performance of a BiDAF model

Answer Length	**F1 (%)**	**EM (%)**	**# Questions**
1-3	73.257	61.805	7,323
4-6	69.004	47.724	1,670
7-9	62.403	33.061	490
10-12	59.746	31.064	235
13+	42.476	9.929	282

4.4 Analysis on long-length predictions

I could notice that the predictions of the start token are relatively accurate than the predictions of the end tokens, and the length of the predictions of the model is longer than the length of the target answers in many cases. The predictions of both the start and the end tokens are made from the output of the final fully-connected layer. Since each Softmax classifier independently predicts the position, the two Softmax layers do not share their prediction results. If we add additional layers to inform the later Softmax layer about the prediction result of the start token, we might reduce this problem.

Examples:

Answer: phlogiston theory
Prediction: phlogiston theory of combustion and corrosion

Answer: reliance on teaching fellows
Prediction: reliance on teaching fellows for some aspects of undergraduate education

4.5 Analysis on pure wrong predictions

Even though the attention mechanism I built sometimes does not work well on queries with long answers, it works reasonably well on queries with short answers. This clearly illustrates the necessity for exploring a better attention mechanism. Equipped with more intuitive attention mechanism, and more question-answer pairs with longer answer length, we might improve the performance.

4.6 Analysis on training process

Our model is trained by minimizing the Softmax losses of predicting the start and the end tokens. We can easily recognize both the target answer and the prediction below as plausible answers for the question. However, since the model only considering the start and the end positions of answers in the training data, the model will penalize the wrong prediction of the start token's position. Investigating better designed objective functions, or adding additional loss terms may prevent these undesirable cases.

Examples:

Question: "why were the initial suggestions for a devolved parliament before 1914 shelved?"
Answer: first world war
Prediction: due to the outbreak of the first world war

4.7 Different optimization objective

So far, the deep recurrent neural networks were trained based on the Softmax loss on the predictions of start and end tokens on a passage for a certain query. Even though the Softmax loss is a fairly good objective to train the networks, some researchers started to optimize different objectives, and this leaded to a high performing question-answering model trained by minimizing the convex combination of the Softmax loss and the negative of the F1 score [5]. Inspired by their successful results, I implemented models which were trained by directly minimizing the negative of the F1 score, and I could find out that the balance between the Softmax loss and the negative of the F1 score is important for training. However, the actual performance of those models was on par or slightly inferior than the performance of models trained by the Softmax loss. This can lead us to two possible conclusions: the network structure is not well chosen for directly minimizing the negative of the F1 score, or the Softmax loss itself is proper objective to minimize.

5 Conclusion

In this project, I explored different deep recurrent neural network structures and diverse attention mechanisms to tackle the SQuAD reading comprehension challenge. The best performing model I found is a modified version of the BiDAF with additional self-attention and highway layers, and that single model achieved the F1 76. 302%, and the EM 66.11% on the dev set and **F1 77.176%** and **EM 67.481%** on the test set. The model clearly demonstrated its ability to capture the relationship between a passage and a query with a series of simple attention mechanisms. Also, additional 1-d convolutional layer can mitigate the problem of handling unknown words, and from this, if we use larger pre-trained word embedding, e.g, GloVe 840B 300d, we might expect additional improvement of our model without any modification of our network structure. In addition, highway network, and residual structure contributed to faster convergence and mitigated the unstable training process due to trainable character embedding layer, and improved the performance.

I plan on performing extension work in a number of areas. The first possible extension can be implementation of gated attention methods. We know that attention is a key factor for improving the performance of a question-answering model. Recently, several researchers suggested gated attention mechanisms, or fusion methods

[5][6][7], and showed outstanding performance on the task. Inspired by their successful results, I'd like to investigate more complicated but intuitive attention mechanisms, and a way to train those networks efficiently.

In addition, instead of using the Softmax loss for training our network, I'd like to develop a model which directly optimizing the F1 score. Instead of using the current framework to minimize the negative of the F1 score, I'd like to apply the Reinforcement Learning (RL) frameworks to tackle this problem. Since we can model the negative of the F1 score as the reward for the RL agent's final prediction (action), I'd like to investigate whether the RL framework develop more powerful question answering agent. Since the number of question-answer pairs are limited, I'd like to do some more experiments with different exploration/exploitation algorithms and sample efficient sampling methods.

Acknowledgements

I would like to thank Richard Socher and all TAs for having us such a great learning opportunity. Also, I would like to thank Microsoft for providing computational resources on the Azure.

References

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text" arXiv preprint arXiv:1606.05250, 2016.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory" Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension" arXiv preprint arXiv:1611.01603, 2016.

[4] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation" in EMNLP, vol. 14, pp. 1532–1543, 2014.

[5] M. Hu, Y. Peng, and X. Qiu, "Reinforced Mnemonic Reader for Machine Comprehension" arXiv:1705.02798, 2017.

[6] R. Liu, W. Wei, W. Mao, and M. Chikina, "Phase Conductor on Multi-layered Attentions for Machine Comprehension" arXiv:1710.10504, 2017.

[7] H. Huang, C. Zhu, Y. Shen, and W, Chen, "FusionNet: Fusing via Fully-Aware Attention with Application to Machine Comprehension" arXiv:1711.07341, 2017.

[8] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway Networks" arXiv:1505.00387, 2015.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition" arXiv:1512.03385, 2015.

[10] Y. Kim, "Convolutional Neural Networks for Sentence Classification" arXiv:1408.5882, 2014.

[11] C. Xiong, V. Zhong, R. Socher, "Dynamic Coattention Networks For Question Answering", arXiv:1611.01604v4, 2017.