Question Answering on SQuAD Dataset with BiDAF and Self-Attention

Junwei Yang
Department of Computer Science
Stanford University
junweiy@stanford.edu

Abstract

Machine Comprehension, answering a question about a given context paragraph, requires modeling complex interactions between the context and the query. Recently, related work about attention mechanisms has been successfully investigated to model these interactions. Among these related works, Bi-Directional Attention Flow Attention [2] and Self-Attention [3] demonstrate state of art performance on question answering task. In this paper, I presented a model, which combines Bi-Directional Attention Flow and Self-Attention. Eventually, the model is able to get F1 score 76.9, EM score 67.2 in Stanford Question Answering Dataset (SQuAD).

(*Note: This is not the final score I achieved on the test set, due to codalab is dysfunctional, my final score will be submitted later on gradescope. The scores here are my previous codalab submission on test set.*)

1 Introduction

Question answering has been a very popular research topic over the past few years due to it can be widely used in various applications. In this paper, the work is mainly focused on the Stanford Question Answering Dataset (SQuAD). SQuAD requires answering questions given a context paragraph and the given answer exists in the given paragraph. Moreover, SQuAD requires different forms of logical reasoning to infer the answer. [2]

Rapid progress and competitive state of art methods has been made since release of SQuAD dataset. Two most recent state of art methods, BiDAF and R-Net, the main ideas behind these works are different attention mechanisms, which tried to capture the interactions between context and question, thus better predicting answer based on question-aware context features. Inspired by these two models, I introduced a model, an end-to-end deep learning network for question answering, which hybrid combines these two methods in this paper.

The rest of the paper is organized as following: Section 2 defines the problem; Section 3 gives a brief overview of the recent related work; Section 4 introduces my method and the details of my implementation based on the previous BiDAF and R-Net methods. I also introduce the training details; Section 5 is result and analysis including an ablation study, and error analysis; Finally in Section 6, summary of this project and future work is discussed.

2 Dataset and Problem Definition

SQuAD consists of 100K question-answer pairs and each one is associated with a context paragraph. The paragraphs in SQuAD are taken from Wikipedia. [1] All the answers in SQuAD dataset are a sub-section in the associated paragraph given. This means the prediction made in SQuAD dataset is a “span”, one start position and one end position, in the paragraph given. In the dev and test set, each question is associated with 3 golden answers. Quantitative evaluation is based on F1 and Exact Match (EM) score. F1 is the harmonic mean of precision and recall and EM is a binary metric, which is true if the prediction matches the golden answer completely.

3 Related Work

Previous works in end-to-end machine comprehension and question answering often investigated powerful attention mechanisms to improve the neural network-based models. Neural network-based models demonstrate the effectiveness on the SQuAD dataset. A match-LSTM and pointer network is introduced in 2016 to produce the boundary of the answer. [6] R-Net introduced a self-matching attention, which dynamically refines the passage representation by looking over the whole passage, and aggregating evidence relevant to the current passage word and question, allowing it make full use of the passage information. [3] BiDAF introduces a 2-way attention, which involve both context to question attention and question to context attention. [2]

For the model implemented in this work, BiDAF and Self-Attention is combined to get an end-to-end model, which has a comparable performance with current state of art approaches.

4 Approach

I will first introduce the model architecture implemented based the on the original BiDAF and R-Net papers. The model has a modular structure, which consists of multiple layers:

- **Contextual Embedding Layer** maps each word to a feature vector space using pre-trained GloVE word embedding matrix.
- **Attention Layer** is to model the complex interactions between context and question and produce final context-aware question feature vector. I implemented this layer with a hybrid attention using BiDAF and Self-Attention.
- **Output Layer** predicted the answer span based on the feature vector output of the attention layer. Two options, output layer in BiDAF paper [2] and answer pointer network introduced in R-Net [3], are explored to improve the simple baseline method provided.

4.1 Architecture Details

Figure 1 shows the model architecture. The details are introduced as follows.

1. **Contextual Embedding Layer.** Word embedding layer maps each word to a high-dimensional vector space. The pre-trained word vectors of GloVE are used to obtain the fixed word embeddings of each word. Consider a question $Q = \{w_t^Q\}_{t=1}^m$ and a context passage $P = \{w_t^P\}_{t=1}^n$. Then we first convert the words using their respective word-level embeddings ($\{e_t^Q\}_{t=1}^m$) and ($\{e_t^P\}_{t=1}^n$). We then use a bi-directional Long Short-Term Memory Network to produce new representations $u_1^Q, \dots u_m^Q$ and $u_1^P, \dots u_n^P$ of all words in the question and passage respectively:

$$u_t^Q = BiRNN\ (u_{t-1}^Q, e_t^Q)$$
$$u_t^P = BiRNN\ (u_{t-1}^P, e_t^P)$$

The forward representation and back representations of BiRNN are concatenated together to get u_t^P and u_t^Q. The weights used for both question and context embedding is shared together.

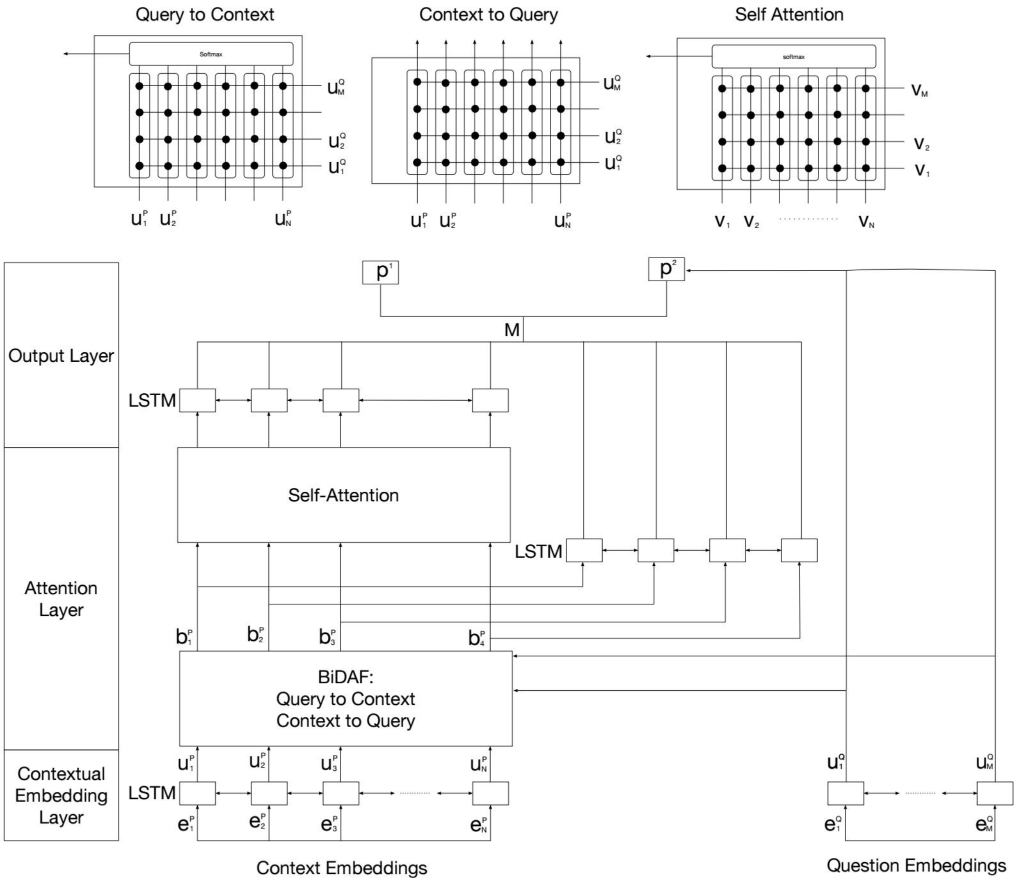

FIGURE 1: MODEL ARCHITECTURES

2. **Attention Layer.** Attention layer is responsible for linking and fusing information from the context and question words. Here I will first introduce two attentions, BiDAF and Self-Attention. The inputs to the layer are the contextual vector representations of the context u_t^Q and question u_t^P.

   BiDAF. The main idea of BiDAF is that attention should flow both ways – from the context to the question and from the question to the context. The details of BiDAF attention layer are explained in the project handout. I followed exactly the same steps in my implementation so I will just skip a detail introduction here.

   Self-Attention. Self-attention is introduced in the R-Net paper. It is as follows. Suppose we have some sequence of representations of $\{v_{1,}v_{2,} \dots v_N\}$, with each corresponding to a context location. Each v_i attends to all the $\{v_{1,}v_{2,} \dots v_N\}$. In equations:

$$e_j^i = v^T \tanh\left(W_1 v_j + W_2 v_i\right) \in R$$

$$\alpha^i = softmax(e^i) \in R^N$$

$$a_i = \sum_{j=1}^{N} \alpha_j^i \, v_j \in R^l$$

   Here W_1, W_2 are weight matrices and v is a weight vector. Then for each context location, concatenate the self-attention output $a_i \, to \, v_i$, this gives the output of self-attention layer.

Additive or Multiplicative. One thing to notice is that the above equations in the self-attention are in the form of additive attention. [5] There is another form of attention, which is multiplicative attention. I experimented both choices in my implementation and measure the performance of system. Multiplicative attention is chosen in my final model. The additive attention cost more memory and due to the GPU memory limit of this work, I can only use a small batch size and small hidden states size for the attention layer. As multiplicative attention is more memory-efficient, thus a larger batch size and a larger hidden states size can be used to improve the performance of the system.

Processing and Fusing of BiDAF and Self-Attention. To get the final output of this attention layer, two remaining steps need to be done. Firstly, process attention output using a Bi-RNN to capture the temporal information in text. Secondly, fuse BiDAF and Self-Attention. Secondly, I tried to fuse both attention output to improve the performance. I used the following sequence of operations to fuse both attentions in my final model:

- (BiDAF + Self-Attention) / (BiDAF + Self-Attention + Bi-RNN)

BiDAF means the BiDAF attention layer, Self-Attention means the Self-Attention layer, Bi-RNN means a bidirectional RNN layer, + sign means taking the output of previous layer as input of next layer, / sign means two layers are parallel and concatenating their output as final output $M \epsilon R^{N \times l}$, where l is the appropriate hidden size.

Note: The BiDAF and Self-Attention here only refer to the corresponding attention layers of the original paper instead of the whole model.

3. **Output Layer.** The output layer is to take the output of the attention layer, referred as M here and predict the answer ‘span’, start location and end location. There are two options investigated based on the paper of BiDAF and R-Net to improve the baseline output layer, which is simply two softmax operations. My final model used the Answer Pointer Network as the output layer.

BiDAF Output Layer. In BiDAF paper, first the probability distribution of the start index over the entire paragraph is computed by

$$p^1 = softmax(wM)$$

$$where\ w\ is\ a\ trainable\ weight\ vector.$$

For the end index of the answer, we pass M to another bidirectional LSTM layer and obtain $M_2 \in R^{N \times l}$. Then we use it to obtain the probability distribution of the end index in a similar manner:

$$p^2 = softmax(wM_2)$$

Thus we can make the end index conditioned on the start index instead of independently in the baseline model.

Answer Pointer Network. Answer point network is a recurrent neural network that can be used to predict the start and end position of the answer. Firstly an attention polling over the question representation is used to generate the initial hidden vector for the pointer network. Suppose the hidden question states are u_i^Q for each question word. The an attention pooling which is similar to Self-Attention is done as follows:

$$e_j^i = v^T \tanh\left(W_1 u_j + W_2 u_i\right) \in R$$

$$\beta^i = softmax(e^i) \in R^N$$

$$a_i = \sum_{j=1}^{N} \beta_j^i v_j \in R^l$$

Then this attention pooling output a_i is used as the initial hidden states h_1 of the answer pointer network. The input of the answer recurrent network is the weighted sum of m_i using the corresponding weights α_i^t:

$$c_t = \sum_{i=1}^{n} \alpha_i^t m_i$$

$$h_t = RNN(h_{t-1}, c_t)$$

Given the output of the attention layer $M \epsilon R^{N \times l}$, another attention is utilized as a pointer to select the start position p^1 and end position p^2 from the passage, which can be formulated as follows:

$$s_j^t = w^T \tanh\left(W_m m_j + W_h h_t\right) \in R$$

$$\alpha^t = softmax(s^t) \in R^N$$

$$p^t = argmax(\alpha_1^t, \dots, \alpha_n^t)$$

4.2 Implementation Details

This subsection gives the implementation details of my final model.

Training Details. I use a learning rate of 0.001 initially. The moving averages of all weights of the model are maintained with the exponential decay rate of 0.999. Batch size 25 is chosen and trained for 25 epochs.

Model Details. For the contextual embedding layer, the dimension of GloVE vector used is 100. I also tried to increase this to 200, but it doesn't increase the performance. The hidden state size for Bi-RNN used is 100 for embedding layer. For the attention layer, the hidden state size used is 75 for both BiDAF and Self-Attention. A dropout rate of 0.2 is chosen.

Ensemble Details. The final ensemble model is trained consisting of 5 training runs with the identical architecture and hyper-parameters. At test time, summing up all the confidence scores of all the 5 models and using the smarter span selection to choose the answer span.

Model	EM	F1
My Implementation (Ensemble)	67.4	76.9
BiDAF (Single Model)	68.0	77.3
BiDAF (Ensemble)	**73.3**	**81.1**
R-Net (Single Model)	68.4	77.5
R-Net (Ensemble)	72.1	79.7

TABLE 1: COMPARISON WITH R-NET AND BIDAF PAPERS

5 Experiments

5.1 Result Analysis

My final model achieved 76.9 F1 score and 67.4 EM score on test set. A detail comparison with state of art original R-Net and BiDAF papers on test dataset is shown in Table 1. From the results, we can see that my model performance is comparable to the state of art question answering methods. The gap between my model and the R-NET and BiDAF papers can be due to that I don't use character level embeddings in the contextual embedding layer. Both BiDAF and R-Net papers use character embeddings, making the model take care of out-of-vocabulary words better.

5.2 Ablations

As mentioned before, the final model is developed incrementally step by step so that I can carefully examine the effects of each technique implemented.

5.2.1 Attention Layer.

The operations to fuse BiDAF attention and Self-Attention have different variants. I tried the following sequence of operations and do an ablation study to better understand the system:

- BiDAF
- Self-Attention + Bi-RNN
- BiDAF + Self-Attention + Bi-RNN
- (BiDAF) / (BiDAF + Self-Attention + Bi-RNN)
- (BiDAF + Self-Attention) / (BiDAF + Self-Attention + Bi-RNN)

Table 2 shows performance results on dev set of all model architectures implemented.

The BiDAF and Self-Attention here only refer to the corresponding attention layers of the original paper instead of the whole model. Bi-RNN means a bidirectional RNN layer, + sign means taking the output of previous layer as input of next layer, / sign means two layers are parallel and concatenating their output as final output $M \epsilon R^{N \times l}$, where l is the appropriate hidden size.

At first, a BiDAF attention layer is added to replace the baseline attention layer. Then I also investigated the model with only Self-Attention. After using single attention separately, I am implementing models by fusing both attentions in different ways as above.

5.2.2 Output Layer

After experimenting all different attentions, I chose the model 5 in Table 2, which perform best, and replaced the simple softmax output layer in baseline with more advanced options. I experimented both the Answer Pointer Network output layer and BiDAF output layer separately.

Method	F1	EM
0. Baseline (Taken from leaderboard trained by Abisee)	43.57	34.67
1. BiDAF + Softmax Output	50.71	40.21
2. Self-Attention + Bi-RNN + Softmax Output	63.94	51.12
3. BiDAF + Self-Attention + Bi-RNN + Softmax Output	66.64	54.78
4. (BiDAF) / (BiDAF + Self-Attention + Bi-RNN) + Softmax Output	68.66	57.04
5. (BiDAF + Bi-RNN) / (BiDAF + Self-Attention + Bi-RNN) + Softmax Output	70.07	59.78
6. (BiDAF + Bi-RNN) / (BiDAF + Self-Attention + Bi-RNN) + BiDAF Output	72.94	62.39
7. (BiDAF + Bi-RNN) / (BiDAF + Self-Attention + Bi-RNN) + Answer Pointer	**73.15**	**63.66**

TABLE 2: ABLATION ON ATTENTION & OUTPUT LAYER

5.2.3 Span Selection.

At test time, I implemented a smarter span selection introduced by DrQA paper [4] by choosing start and end location with $i \leq j \leq i + 15$ that maximize $p_i^1 * p_j^2$. The results are shown in Table 3.

Method	F1	EM
Model 7 w/o Smarter Span	73.15	63.66
Model 7 with Smarter Span	**74.07**	**64.12**

TABLE 3: ABLATION ON SPAN SELECTION

5.2.4 Analysis

For attention layer**,** from the results in Table 1, it can be seen that with only BiDAF attention, the performance is only improved ~7% on the baseline model, this is from the additional question to context layer. With only self-attention and a Bi-RNN layer, the performance is boosted about ~20% above the baseline. It can be seen that Bi-RNN is very important in the model to capture the temporal information in the context and question. The final model process BiDAF using another Bi-RNN layer in parallel to the processing of Self-Attention features, and concatenating both output features together as the final output of attention layer. For the output layer, either BiDAF Paper output layer or Answer Pointer Network gives us about 3% boost.

5.3 Error Analysis

The errors made in the dev set by our model are analyzed and categorized. I randomly selected 50 (question, context, answer) tuples, which are not predicted correctly based on F1 threshold with 75% and analyzed them.

5.3.1 Imprecise boundaries

42% of the errors made are falling into this category. The precise boundaries are very hard to capture. Also by looking at the examples carefully, I do not consider some of these as "errors".

- **Context**: The European Commission is the main executive body of the European Union. Article 17(1) of the Treaty on European Union states the Commission should "promote the general interest of the Union" while Article 17(3) adds that Commissioners should be "completely independent" and not "take instructions from any Government". Under article 17(2), "Union legislative acts may only be adopted on the basis of a Commission proposal, except where the Treaties provide otherwise." This means that the Commission has a monopoly on initiating the legislative procedure, although the Council is the "de facto catalyst of many legislative initiatives". The Parliament can also formally request the Commission to submit a legislative proposal but the Commission can reject such a suggestion, giving reasons. The Commission's President (currently an ex-Luxembourg Prime Minister, Jean-Claude Juncker) sets the agenda for the EU's work. Decisions are taken by a simple majority vote, usually through a "written procedure" of circulating the proposals and adopting if there are no objections.[citation needed] Since Ireland refused to consent to changes in the Treaty of Lisbon 2007, there remains one Commissioner for each of the 28 member states, including the President and the High Representative for Foreign and Security Policy (currently Federica Mogherini). The Commissioners (and most importantly, the portfolios they will hold) are bargained over intensively by the member states. The Commissioners, as a block, are then subject to a qualified majority vote of the Council to approve, and majority approval of the Parliament. The proposal to make the Commissioners be drawn from the elected Parliament, was not adopted in the Treaty of Lisbon. This means Commissioners are, through the appointment process, the unelected subordinates of member state governments.
- **Question**: How are decisions made on behave of the EU made?
- **Prediction**: decisions are taken by a simple majority vote
- **Answer**: ['simple majority vote', 'simple majority vote', 'a simple majority vote']

This prediction is actually right when we look at that as a human. In order to predict the exact boundaries, first is to improve the models to understand the question and answer more deeply. Another important thing is to make sure the data is annotated very consistently as different persons might have different answers.

5.3.2 Syntactic complications and ambiguities

In the following example, the question is asking "what", but the answer is actually a person, which is more appropriate for question starting with "who". This syntactic ambiguity makes the model hard to predict the right answer. One thing to improve this situation is to add additional input features to the model, such as part-of-speech tags, which is demonstrated to be very useful in the DrQA paper. [4]

- **Context:** CBS broadcast Super Bowl 50 in the U.S., and charged an average of $5 million for a 30-second commercial during the game. The Super Bowl 50 halftime show was headlined by the British rock group Coldplay with special guest performers Beyoncé and Bruno Mars, who headlined the Super Bowl XLVII and Super Bowl XLVIII halftime shows, respectively. It was the third-most watched U.S. broadcast ever.
- **Question:** What halftime performer previously headlined Super Bowl XLVII?
- **Prediction:** super bowl 50
- **Answer: [‘**Beyoncé’, ‘Beyoncé’, ‘Beyoncé’]

5.3.3 Answer Missed Completely

There are some cases the answer is completely wrong. For the following example, in the question there’s word ‘Panthers’ and word ‘practice’. Then the model puts more attentions on ‘practice’ and focus on the second sentence of the context, thus giving the wrong answer. In SQuAD, the passage consists of several sentences and the answer span always falls into one sentence. It might be better to consider a sentence ranking approach before predicting the answer. One way is to train a sentence-ranking model separately and combine this with the span prediction model. The other way is to make an end-to-end model considering both cues.

- **Context**: The Panthers used the San Jose State practice facility and stayed at the San Jose Marriott. The Broncos practiced at Stanford University and stayed at the Santa Clara Marriott.
- **Question**: Where did the Panthers practice at for Super Bowl 50?
- **Prediction**: stanford university
- **Answer**: [‘San Jose State practice facility’, ‘the San Jose State practice facility’, ‘San Jose State’

6 Conclusion

In this paper, I implemented a question answering system evaluated on Stanford Question Answering dataset, which combines two state-of-art approaches, BiDAF and R-Net. The performance of my implemented model has results comparable with the current state of art approaches. The ablation study shows that by combining two approaches jointly, the performance is better than just using a single attention approach.

By analyzing the results and errors, there are multiple improvements that can be considered to improve the performance even better. Firstly, character level embedding can be added to improve the results as the original BiDAF and R-Net paper did. Secondly, additional input token features, such as part-of-speech tags and named entity recognition, can be added in the contextual embedding layer to deal with the syntactic ambiguities. Thirdly, as in SQuAD dataset, the answer always span only in 1 sentence. We can train a model can first involve a task of sentence ranking and then select the span in that sentence, thus improving the performance. Finally, to make the boundary more precise, a multiple-hop inference can be introduced in the output layer.

Acknowledgments

I would like to thank the entire course staff for giving great lectures, preparing this great project and timely answering question through Piazza and office hours. I also want to thank all the classmates by giving great ideas and questions through Piazza.

References

[1] Rajpurkar P, Zhang J, Lopyrev K, Liang P (2016). SQuAD: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*

[2] Seo M, Kembhavi A, Farhadi A, Hajishirzi H (2017). Bidirectional Attention Flow for Machine Comprehension. *arXiv preprint arXiv:1611.01603*

[3] Natural Language Computing Group, Microsoft Reserch Asia (2017). R-Net: Machine Reading Comprehension with Self-Matching Networks.

[4] Chen D, Fisch, A, Weston J, Bordes A (2017). Reading Wikipedia to Answer Open-Doman Questions. *arXiv preprint arXiv:1704:00051*

[5] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A, Kaiser L (2017). Attention is All You Need. *arXiv preprint arXiv:1706:03762*

[6] Wang S, Jiang J (2016). Machine Comprehension using Match-LSTM and Answer Pointer. *arXiv preprint arXiv:1608:06905*