Reading Comprehension on the SQuAD Dataset

Zheqing (Bill) Zhu *
Department of Management Science and Engineering, Stanford University
Facebook Inc.
zheqzhu@stanford.edu / billzhu@fb.com

David Xue *
Department of Computer Science, Stanford University
dxue@cs.stanford.edu

Abstract

In this project, we explore various number of established approaches including Bidirectional Attention Flow (BiDAF) and R-Net on SQuAD and some ensembles combining these two approaches in multiple ways. Our model achieved a final performance of 75.017% F1 and 65.027% EM on the test set with BiDAF combined with Self Attention where Dynamic Answer Pointer Programming is implemented.

1 Introduction

Question-answering artificial intelligence has been a popular machine learning problem in the recent few years. In order to uniformly tackle this problem, the Stanford Question and Answering Dataset (SQuAD) was invented to provide a standard platform for innovations in machine comprehension technologies. In this paper, we combine several approaches to tackle SQuAD including the state-of-the-art R-Net and Bidirectional Attention Flow (BiDAF).

In Section 2, we introduce the definition of the SQuAD problem. In Section 3, we describe our dataset and summarize some baseline statistics. In Section 4, we describe the models we implemented for this project with detailed descriptions in neural network layers and architectures we designed. In Section 5, we present the result and the analysis.

2 Problem Definition

Given a sequence of word vectors $c = c_1, c_2, ..., c_N$ and a sequence of word vectors $q = q_1, q_2, ..., q_N$, we aim to find a mapping $(c, q) \mapsto (p_s, p_e)$ where p_s is the start position of the answer and p_e is the end position. The goal is to maximize the similarity between human derived answers and machine derived answers by looking at F1 and EM metrics. F1 metric is defined as the harmonic mean of precision and recall of the machine prediction and EM (Exact Match) is a binary measure of whether the prediction is exactly equal to the ground truth.

3 Dataset

The dataset for this task is the Stanford Question Answering Dataset (SQuAD), a reading comprehension dataset built by crowdworkers on a set of Wikipedia articles. The dataset contains more than 100,000 question-answer pairs on 500+ articles. The data is split into 80% training set, 10%

*Joint First Author

development set, and 10% test set (hidden). 5% of the training set is taken for validation set, which is used in hyperparameter search.

The histograms of the question, context, and answer length of the training set can be seen below in Figure 1. Using the histogram, we set the maximum length of the context and question to 400 and 30, respectively. Setting the limit of context and question drastically reduces the number of parameters and helps with training time. To validate this idea, we examined the 10391 context examples in the dev set and we only found one example where the actual answer end span extended beyond our cutoff of 400.

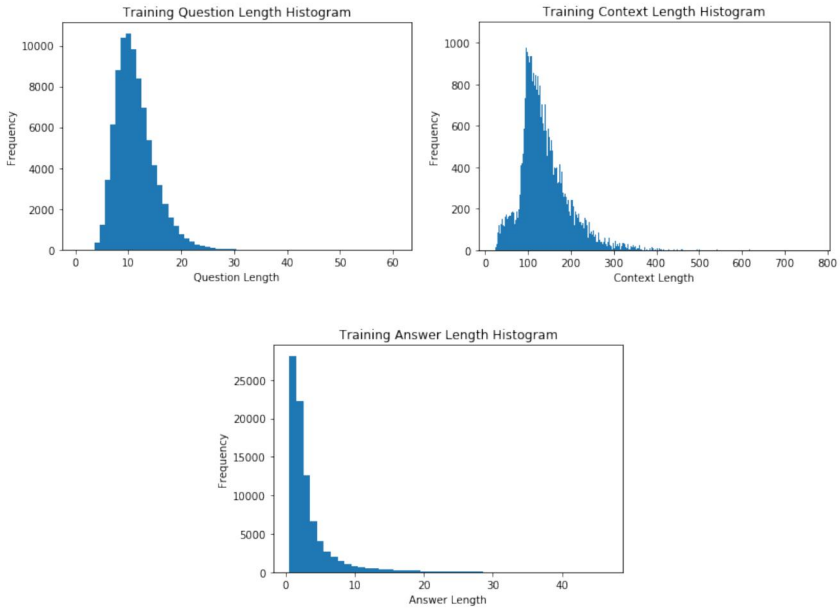


Figure 1: The histograms of the question, context, and answer length of the training set.

4 Model

4.1 Layers

Embedding layer: We found that increasing our word embedding size from 100 to 200 or 300 did not lead to any improvement in the performance of our model, so we finalized the embedding size to 100.

Gated Attention-Based Recurrent Neural Network [1, 3]: The Gated Attention-Based Recurrent Neural Network is used to determine the importance of information in the passage regarding to a question. The architecture is as the following:

$$
\begin{aligned}
v_t^P &= \text{BiRNN}(v_{t-1}^P, c_t) \\
s_j^t &= v^T tanh(W_u^Q u_j^Q + W_u^P u_t^P + W_v^P v_{t-1}^P) \\
a_i^t &= softmax(s_i^t) \\
c_t &= \sum_{i=1}^{m} a_i^t u_i^Q \\
v_t^P &= \text{BiRNN}(v_{t-1}^P, [u_t^P] \\
g_t &= sigmoid(W_g[u_t^P, c_t]) \\
[u_t^P, c_t]^* &= g_t \odot [u_t^P, c_t]
\end{aligned}
\tag{1}
$$

The output of Gated Attention-Based RNN is $[u_t^P, c_t]^*$, which encodes the relationship between the question and the current passage word. The representation generated from Gated Attention-Based RNN can pinpoint important parts of the context. Note that the original R-Net paper did not explicitly indicate the use of BiRNN, but their implementation exploits BiRNN to encode representations. The input of this layer is the encoded representation from the question and the context (u^Q and u^P).

Self-Matching Attention [1, 3]: In order to incorporate the overall knowledge of the context with the representation of Gated Attention-Based RNN output, it is important to encode context awareness in the representation to infer the answer to questions. Hence, Self-Matching Attention is used to match pinpointed representation with overall context. The general layer architecture is the following:

$$\begin{aligned} h_t^P &= \text{BiRNN}(h_{t-1}^P, [v_t^P, c_t]) \\ s_j^t &= v^T tanh(W_v^P v_j^P + W_v^{\bar{P}} v_t^P) \\ a_i^t &= softmax(s_i^t) \\ c_t &= \sum_{i=1}^{n} a_i^t v_i^P \end{aligned} \tag{2}$$

Note that W_v^P and $W_v^{\bar{P}}$ are two different weight matrics and the input of this layer is a blended representation from last layer.

Bidirectional Attention Flow (BiDAF) [2]: BiDAF is used to link and fuse information from the context and the query words. The core concept which led to the win is that attention should flow both from the context to the question and from the question to the context. Fist we compute the similarity matrix:

$$S_{ij} = w_{sim}^T [c_i; q_j; c_i \odot q_j] \tag{3}$$

where c_i and q_j are a context hidden state and a question hidden state respectively. Then we perform Context-to-Question Attention and Question-to-Context Attention:

1. Context-to-Question Attention (C2Q):

$$\begin{aligned} \alpha^i &= softmax(S_{i,:}) \\ a_i &= \sum_{j=1}^{M} \alpha_j^i q_j \end{aligned} \tag{4}$$

where a is output.

2. Question-to-Context Attention (Q2C):

$$\begin{aligned} m_i &= max_j(S_{ij}) \\ \beta &= softmax(m) \\ c' &= \sum_{i=1}^{N} \beta_i c_i \end{aligned} \tag{5}$$

where c' is the output. Note that c' is different from the original context c.

Finally, the output of BiDAF is

$$[c_i; a_i; c_i \odot a_i; c_i \odot c']. \tag{6}$$

Dynamic Programming: After getting the logits for start and end positions, we use dynamic programming strategies for the final answer span prediction. In the baseline model, the answer span (s, e) was selected with independent maximum probability p_s, p_e. In figure 2 we see the end position lies before the start position, but we should never choose an end position before a start position. Therefore, we select (s, e) with the maximum value of p_s^1 p_e^2 where $s \leq e$, which can be calculated in linear time using dynamic programming. In this way, we are able to select among all possible answer spans where the joint probability of start and end are the highest. After using this answer span selection strategy, the F1 and EM scores improved by 2.2 points and 1.3 points, respectively, on the test data set.

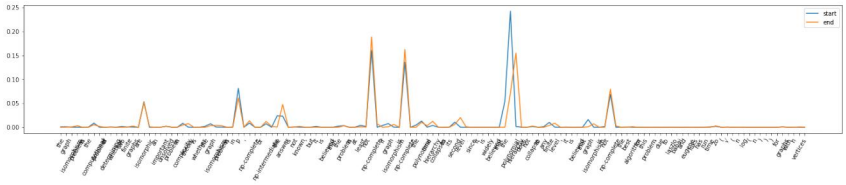
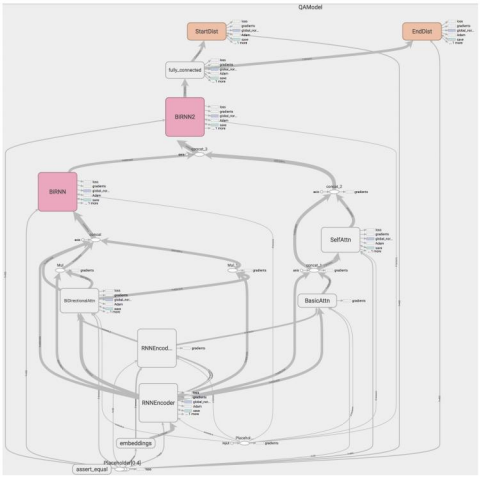


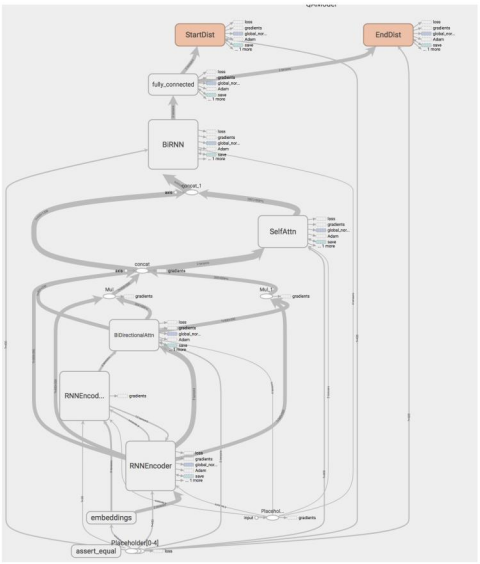
Figure 2: The independent maximum likelihood start position can occur after the independent maximum likelihood end position.

4.2 Architecture

We implemented 5 different architectures to improve the predicting performance as the following (the global hidden state size is 100):



(a) Architecture of BiSelfAttn



(b) Architecture of BiDAF2Self

Figure 3: Architectures of BiDAF2Self and BiSelfAttn

Self-Attention: Self-Attention is a model to improve on top of the existing baseline model by adding a Self-Matching Attention layer after the basic attention layer. Our experiments show that the size of the weight vector does not impact the model performance too much.

BiDAF: Our BiDAF implementation directly mimics the original paper.

R-Net: Our implementation of R-Net only includes the Gated Attention-Based RNN layer and the Self-Matching Attention layer. The Point Network output layer is omitted from our implementation due to the extremely slow convergence of R-Net.

BiDAF2Self: The BiDAF to Self architecture is implemented such that the basic attention layer is replaced by BiDAF and fed into a Self-Matching Attention Layer. Please see Figure3 for more details. The reason why we feed BiDAF results to Self-Matching Attention is to make sure that both C2Q and Q2C representations are encoded with the global context in order to infer the exact location of the answer.

BiSelfAttn: The Bidirectional + Self Attention architecture is used to incorporate the core concepts from both BiDAF and R-Net. We use a BiDAF layer to get a blended C2Q and Q2C representation and use a Self-Matching Attention Layer following the basic attention layer to build a context-aware representation. Following that, we concatenate both representations and feed them into the baseline

output layer with dynamic programming to ensure feasibility of start point and end point. Please see Figure3 for more details. Here we create an ensemble of models by concatenating context-aware representations with C2Q and Q2C combined representation. The rationale is similar to ensembling where by using two forms of attention we are essentially enhancing the representational capacity and flexibility of the model.

4.3 Training Details

Optimizer: After various several learning rates, we used Adam optimizer with an initial learning rate of 0.001.

Dropout: We applied weight decay with a rate of 0.999 and dropout to all layers with a rate of 0.2. This drastically helped with overfitting, as can be seen in Figure 4. However we still observe a 10 point F1 score gap between training and validation.

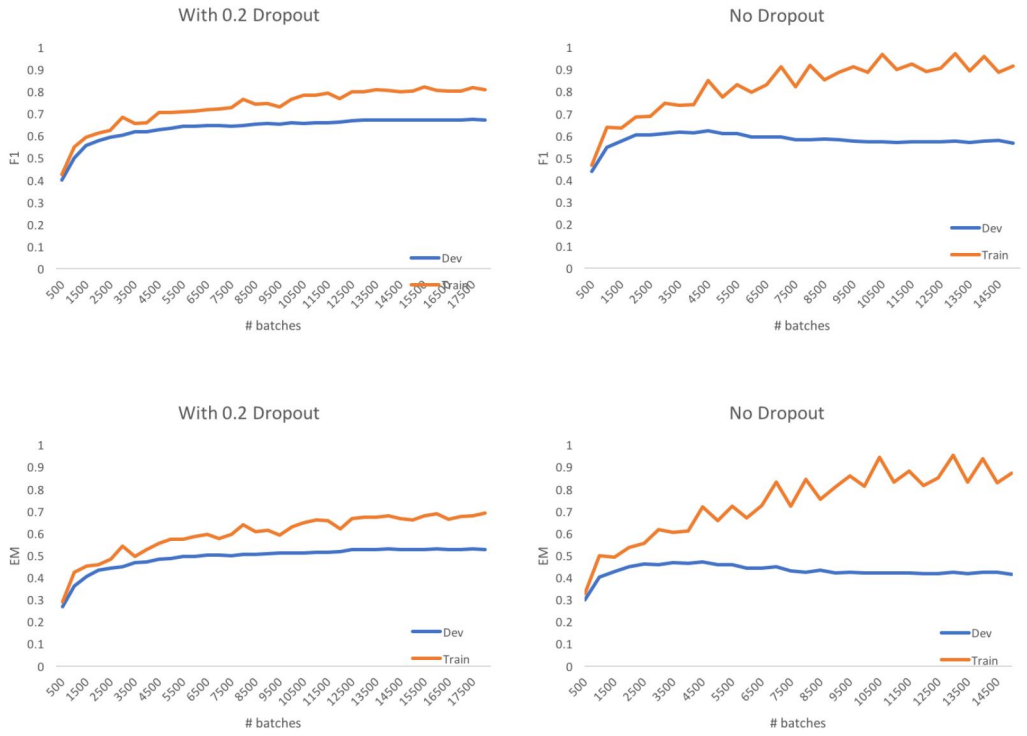


Figure 4: F1 and EM performance of our model with and without dropout.

5 Results and Analysis

In this project, we explore various number of established approaches including Bidirectional Attention Flow (BiDAF) and R-Net on SQuAD and some ensembles combining these two approaches in multiple ways. Our model achieved a final performance of 75.017% F1 and 65.027% EM on the test set with BiDAF combined with Self Attention where Dynamic Answer Pointer Programming is implemented.

5.1 Performance Analysis

Here we compare our results with the original BiDAF and R-net on both Dev set and Test set, see Table 1.

Architecture	F1 & EM (Dev)	F1 & EM (Test)
Self-Attn	65.42, 50.82	
BiDAF (our implementation)	60.86, 47.03	
R-Net (our implementation)	52.85, 40.45	
BiDAF2Self	59.17, 44.45	
BiSelfAttn	67.14, 52.75	73.198, 63.653
BiSelfAttn with DP	67.14, 52.75	75.017, 65.027
BiDAF (original)		77.323, 67.974
R-Net (original)		84.265, 76.461

Table 1: Performance Analysis

Due to submission limitations on CodaLab, the Dev scores in this table are final Dev set scores presented by TensorBoard (which is a very different from the Dev scores on the Dev leaderboard). For example, BiSelfAttn with DP achieves F1 of 74.075 and EM of 63.548. One important note here is that the two BiSelfAttn models (with and without DP) are presented with the same score on Dev because they are trained in a single sequence as DP does not affect training results.

From the results above, we can see that BiSelfAttn overall outperforms all individual single models as an ensemble, which indicates that concatenating BiDAF output and Self-Matching Attention output provides richer information to the output layer (by comparing BiSelfAttn and BiDAF2Self). It also gives a takeaway that more information is better than a more complex embedding.

We also noticed in our implementation, R-Net performs very poorly compared to the original paper and the global leader board. There are two possible reason for that. One possibility is that we did not implement the output point network. The other is that R-Net converges extremely slow due to its heavy variable load. We plan to further investigate these possibilities in the future.

Finally, we observed that most of the original experiments by other researchers perform better than our implementation, which could very much be parameter tuning in dropout rate, hidden state size and other variable parameters. We also plan to explore more possibilities in that space.

5.2 Error Analysis

Below we examine examples from two major errors classes that our model incorrectly classifies in the dev set and provide possible solutions to address them.

5.2.1 Imprecise Span Selection due to Question Ambiguity

Many of the misclassifications of our model slightly extend beyond the tail or start before the head of the true answer span due to ambiguities in the question. In the examples below, the questions themselves are ambiguous about the amount of detail required in the answer. In an example below, our model predicted ”7 January 1900” while the truth is just ”1900”. However both should be considered valid answers. This is a similar case for ”Pac-12” and ”Pac-12 conference” as well as ”high school” and high school level”. This problem harms the performance of our model since most answers are quite short in the SQuAD dataset (50% answers are 10 words or fewer). This illustrates some inherent difficulties with building any question answering system.

1. College sports are also popular in Southern California. The UCLA bruins and the USC Trojans both field teams in NCAA Division I in the Pac-12 conference, and there is a longtime rivalry between the schools.
   QUESTION: Which conference do the teams in Southern California play in?
   TRUE ANSWER: Pac-12
   PREDICTED ANSWER: Pac-12 conference
2. Rugby is also a growing sport in Southern California, particularly at the high school level, with increasing numbers of schools adding rugby as an official school sport
   CONTEXT: At which level of education is this sport becoming more popular?
   TRUE ANSWER: high school

PREDICTED ANSWER: high school level

3. On 7 January 1900, Tesla left Colorado springs. [citation needed] His lab was torn down in 1904, and its contents were sold two years later to satisfy a debt.
   QUESTION: When did Tesla depart from Colorado Springs ?
   TRUE ANSWER: 1900
   PREDICTED ANSWER: 7 January 1900

It could be useful to add an additional LSTM layer before selecting the start and end to help decide on the answer span boundary. Additionally choosing smarter selection span mechanisms than our dynamic programming strategy at the word-level could be useful including penalizing tail-end redundant words in the answer (like in Example 1 and 2) as well as testing other heuristics for predicted answer length (e.g. shorter questions and shorter answers).

5.2.2 Incorrect Attention

In some misclassifications, our model pays attention to the wrong date or part of the context. In the first example below, there are many dates in the passage and our model incorrectly selected 1985. Our model incorrectly pays attention to "the computational model" in the second second example instead of "hides constant factors and smaller terms". In the third example our model pays attention to the technically correct but indirect answer span as opposed to "new entrants to the teaching profession" which is directly next to the answer. These misclassifications have a large penalty on the F1 and EM score due to completely mismatched answer spans.

1. On May 21, 2013, NFL owners at their spring meetings in Boston voted and awarded the game to Levi's stadium. the $1.2 billion stadium opened in 2014. it is the first super bowl held in the San Francisco Bay Area since Super Bowl XIX in 1985, and the first in California since Super Bowl XXXVII took place in San Diego in 2003 .
   QUESTION: When was Levi's stadium picked for Super Bowl 50?
   TRUE ANSWER: May 21 , 2013
   PREDICTED ANSWER: 1985
2. Upper and lower bounds are usually stated using the big O notation, which hides constant factors and smaller terms. This makes the bounds independent of the specific details of the computational model used. For instance, if t(n) = 7n2 + 15n + 40, in Big O notation one would write t(n) = o(n2).
   QUESTION: Big O notation provides autonomy to upper and lower bounds with relationship to what?
   TRUE ANSWER: the computational model
   PREDICTED ANSWER: hides constant factors and smaller terms
3. From 2006 garda vetting has been introduced for new entrants to the teaching profession. These procedures apply to teaching and also to non-teaching posts and those who refuse vetting "can not be appointed or engaged by the school in any capacity including in a voluntary role". Existing staff will be vetted on a phased basis.
   QUESTION: Who is subject to vetting ?
   TRUE ANSWER: new entrants to the teaching profession
   PREDICTED ANSWER: non-teaching posts and those who refuse vetting "can not be appointed or engaged by the school in any capacity including in a voluntary role

It could be helpful to add more layers of attention to compute the context-to-query (C2Q) and query-to-context (Q2C) attention. The motivation for deeper attention layers is that attention signal can flow through different regions with multi-step understandings and our model can build a more complex and comprehensive understanding of relationship between the context and query.

Acknowledgments

The authors would like to thank the TAs for their support and guidance during this project. We would also like to thank Microsoft for providing the GPU Virtual Machines on Azure to train the models.

References

[1] M. R. A. Natural Language Computing Group. R-net: Machine reading comprehension with self-matching networks. 2018.

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

[3] W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198, 2017.