Question Answering with Attention

Sajana Weerawardhena

1

{sajana}@stanford.edu.br

Abstract. *Analysis of the use of a BiDAF model on the SQuAD*
***Keywords**: BiDAF, SQuAD.*

1. Introduction

Effective, flexible models for reading comprehension has long been a goal of Artificial Intelligence. While on one hand seen as a necessary prerequisite for General AI, it is also valuable commercially especially in customer communications. In 2016, in an effort to further this field of study, the Stanford NLP group (Rajpurkar et al 2016) released Stanford Question Answering Dataset (SQuAD) which is reading comprehension dataset drawing from Wikipedia Articles, consisting of 100,000+ question-answer pairs on 500+ articles. SQuAD is significantly larger than previous reading comprehension datasets. The main evaluation metrics for SQuAD are the F1 score (based off Recall and Precision) and the Exact Match (EM) score. In this paper, we will discuss the use of a Bidirectional Attention Flow model (BiDAF) on SQuAD.

2. Prior Work

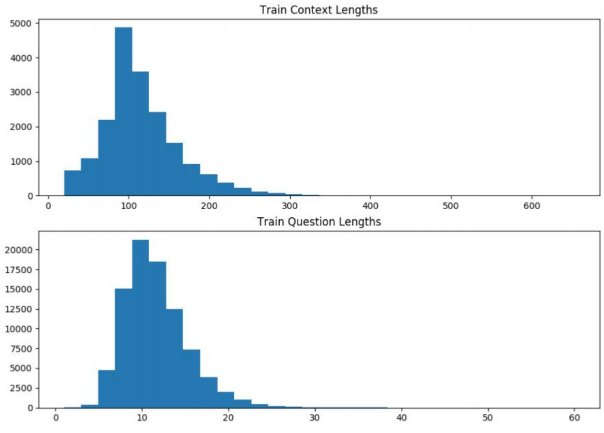
Numerous organizations have made significant progress on this dataset. SQuAD maintains a leaderboard for the best models. Throughout the leaderboard, there are repeat references to different components of models. BiDAF is one such high performing model which was initially described by Minjoon Seo et al. There are numerous models in the top performing models on the leaderboard that use BiDAF at some level. Often BiDAF is combined with iterative self attention, Elmo and so on. Most models that include BiDAF have F1 scores greater than 77, and EM scores greater than 66. Other models that have been repeated include the Dynamic Coattention model as described by Caiming Xiong, Victor Zhong, Richard Socher in their paper Dynamic Coattention Networks for Question Answering, ICLR 2017 and the Reinforced Mnemonic Reader as seen in Reinforced Mnemonic Reader for Machine Comprehension by MinghaoHu, Yuxing Peng, Xipeng Qiu. Both these implementations seemed to perform better than models only using BiDAF.

While EM/F1 Scores have been priority for the aforementioned implementations, another field of study is exclusively focused on developing memory-efficient models or time-efficient models for this dataset. One aspect of this is to develop a Semi-supervised model for SQuAD by reducing the size of the training set.

After combing through these implementations, we decided we wanted to focus on solely optimizing our F1 and EM scores. Balancing reasonable F1/EM scores and a feasible starting point for our project, we decided to pursue a BiDAF implementation.

3. Preprocessing

Most of the prepossessing we did was involved in understanding the lengths of the contexts and questions.



As seen above, we noticed that while context words were mostly below 300 in length, there was still a couple of contexts with over 300. Since the model is going to pad each context vector to the maximum context vector size, we chose to cap the contexts at length 300. Through similar deduction we chose to limit the question lengths to 20 as well.These changes would allow us to train faster and more effectively.

4. Baseline

The baseline we implemented had 3 layers. The first layer is an RNN encoder layer, 2 LSTM Cells, which takes in the question and context vectors in the form of word embeddings from pretrained GloVE vectors.

The second layer was a bidirectional attention. The attention however had a slight modification: the similarity function was only based off the element-wise multiplication of the context vectors and question vectors.

$$S_{i.j} = w^T sim[c_i \odot q_j] \in R$$

Instead of

$$S_{i.j} = w^T sim[c_i; q_j; c_i \odot q_j] \in R$$

The final layer is a fully connected layer that feeds into pair of softmax activations, that take in the attention output and the context vectors and then independently outputs the start vector and the end vector. The loss was the sum of the cross entropies for the start and end index.

5. Bidirectional Attention Model

We implemented the Bidirectional Attention Flow described in Bidirectional Attention Flow for Machine Comprehension (2016) by Minjoon Seo et al, without the Character-level Convolutional Neural Network. The version we implemented had 4 layers. The first two layers were the same as the baseline except the bidirectional attention had the original similarity function:

$$S_{i.j} = w^T sim[c_i; q_j; c_i \odot q_j] \in R$$

The next level was a modelling layer which is composed of 2 LSTMS.

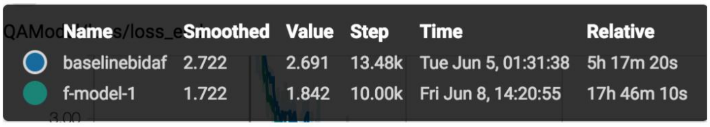
The final layer, the loss and the learning algorithm was also the same as the baseline

6. Architecture Search and Hyperparameter Search

The project was an exercise in Architecture and hyperparameter search. We first decided to improve on the basic attention in the default starter code (which was a dot product of the context and question vectors). We did this by implementing a BiDAF- but since our final goal was a BiDAF, for the baseline we decided to experiment with a different similarity function. The baseline did not have any regularization or dropout implemented so it overfitted a lot. For the next model, we added the original Bidirectional attention and a modelling layer with dropout and regularization at every layer. This performed far better than the baseline.

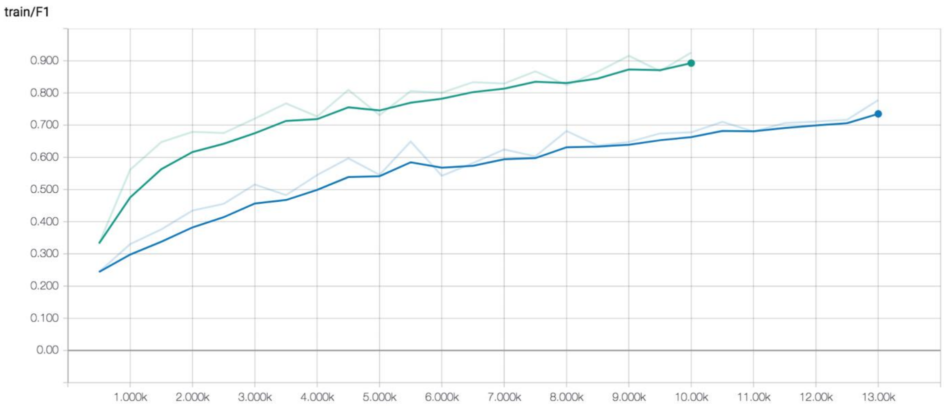
7. Results

On the Dev set, after 15 hours of training and 10,000 iterations, our final model scored 66 F1 and 51 EM.
During training, we noticed that our final model took so much longer to train that our baseline as seen in the figure below.



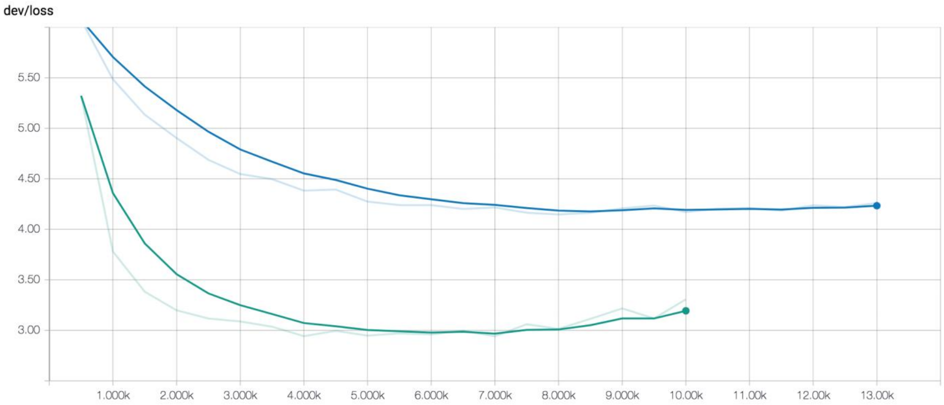
	Name	Smoothed	Value	Step	Time	Relative
●	baselinebidaf	2.722	2.691	13.48k	Tue Jun 5, 01:31:38	5h 17m 20s
●	f-model-1	1.722	1.842	10.00k	Fri Jun 8, 14:20:55	17h 46m 10s

The train loss showed a steady downward trend even after 10k iterations for the final BiDAF model. The train loss for the baseline had plateaued at this point (see Appendix A). Both the train EM and train F1 also showed a continued upward trend even after 10k iterations. We present only the graph for train F1 here (see Appendix A for EM). Train F1 peaked at 0.9253 and train EM peaked at 0.8390.



The Dev BiDAF F1 was over 20 points above the baseline, peaking at 0.66. Similarly the BiDAF F1 was over 20 points above the baseline, peaking at 0.66.

The dev loss was lowest at around 5000 to 6000 iterations, where it touched 2.948 Loss but by 10,000 iterations the loss had climbed to 3.306. This shows that despite dropout and regularization, our model still overfitted to the train.



The large difference in the dev F1 and EM and the train F1 and EM points to some underlying overfitting in the model. However, when we look at the train loss and the dev loss, at their closest they were just 0.3 apart (train loss of 2.691 vs dev loss of 3.306). This points to EM and F1 scores not having a close correspondence with the loss.

We could not run it on the test set because this involves submitting to the leaderboard. On the test set, the leading model has scored 83.877 EM and 89.737 F1. The human benchmark for this set is 82.304 and 91.221 F1. So our model has still a long way to go.

8. Analysis

In order to understand how well the model was doing, we decided to analyse a sample of the dev set (70 samples) that it was evaluated on. The entire SQuAD dataset has 107,785 question-answer pairs: only 10% of that was set aside for the dev set. Hence 70 samples accounts for only 0.0064% of the dev set. Therefore we have to be cautious about any generalizations we make through our observations.

As seen below we categorized the examples by question type (how, why, what, who, where and when, hence H5W). By looking at the samples through this lens we were able to get a granular view of how our model performs on different question types.

Question Type	Percentage	Example
Who	14.2%	-
Where	15.7%	-
What	41.42%	-
When	15.71%	-
Why	1.42%	-
How	5.71%	-
Other	5.71%	Name a ; Which person

Figure 1: Question Distribution in Sample.

Overall of the 70 samples, 58% of the samples had a predicted answer that was an exact match to the ground truth answer.These splits differed drastically within each of the H5W classes as well. Our model fared well against when/where type of questions but was challenged under what/how questions. Therefore we will analyze different groups of H5W classes.

Our original analysis goes into depth about the differences between these classes, quoting examples of each. This analysis has been moved to the Appendix, in order to facilitate the 5 page limit on this paper. Below, we aim to summarize the key findings for each class.

8.1. ”What”Types of Questions

Firstly we noticed that ”what”questions had a diverse range of question patterns (eg - What X thought of Y, What does Y, What profession ...) and they also draw from a range of different answer types (Noun/Adjective/verb phrases, clauses, locations etc). We found this represented a considerable challenge to our model. Overall of the 29 samples had a 50 EM score and an average F1 score of 0.553.

Next, we noticed a lot of the errors were ”off by a couple words”errors: intuitively most required words were returned by model but also some other words too. This causes the F1 score to suffer because certain words in the answer are not more weighted than others. The choice of shorter answers over longer ones, is an arbitrary one but also one that makes sense. However we found examples in the dev set that supported shorter answers and others that supported longer answers over shorter ones. An example of the latter is seen here:

> **Context:**
> by 1998 , the UNK had grown to connect more than 100 universities and research and engineering institutions via 12 national points of presence with UNK (45 mbit/s) , UNK (155 mbit/s) , and UNK (622 mbit/s) links on an all UNK backbone , a substantial engineering feat for that time . the UNK installed one of the first ever production UNK (2.5 gbit/s) ip links in february 1999 and went on to upgrade the entire backbone to UNK .
> **Question:** by 199 how many universities were connected
> **True answer:** by 1998 , the vbns had grown to connect more than 100 universities and research and engineering institutions via 12 national points of presence with ds-3
> **Predicted Answer:** 100

Notice the difference between the predicted and true answers. Given this variety was found in a microcosm of the larger data-set, it is possible that there are lot of training examples that have similar contradictions. As a result the model cannot learn to favour smaller answers over longer ones (if this was the goal).

8.2. When, Where and Who

When, Where and Who questions generally performed really well. When questions had an average EM score of 90%; Where had 63%; Who had 70%. We think this is largely because answers to these types of questions are a very specific type (location, date, number etc) and it is relatively easier for the model to learn to filter for that type based on the question using word embeddings.

9. Conclusion and Future Work

9.1. Conclusion

Overall, our modelling layer and similarity function significantly improved on the baseline. However we learned that there are still barriers for the model in the form of what/how type questions, especially because of off by a few words errors.

9.1.1. Future Work

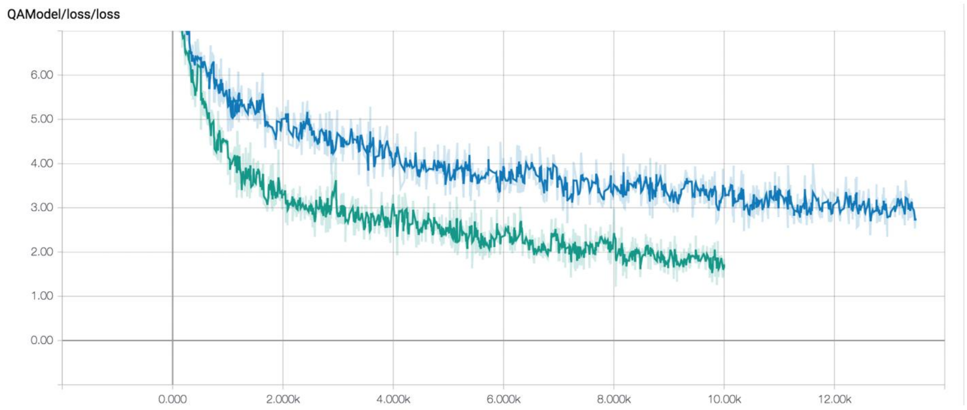
We hope to expand on our work by implementing a Character CNN in order to augment our embeddings in the model. We also wanted to implement an attention visualizer to inform our sample analysis but could not do so, due to time constraints. We are looking into following up on this commitment as part of our future work.

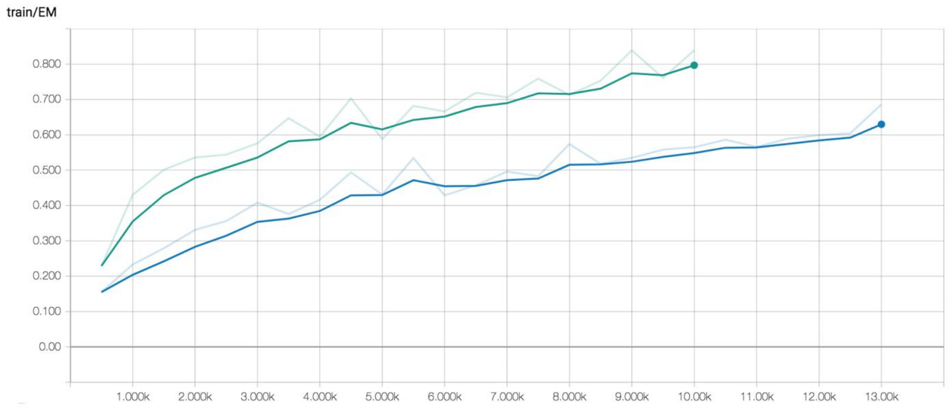
10. Worked Cited

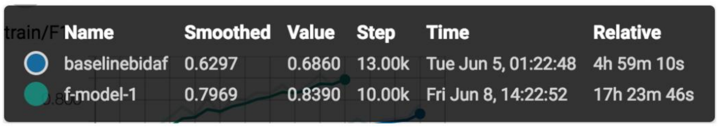
Rajpurkar, Pranav Zhang, Jian Lopyrev, Konstantin Liang, Percy. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text. 2383-2392. 10.18653/v1/D16-1264.
Machine Comprehension Using Match-LSTM and Answer Pointer Shuohang Wang, Jing Jiang: https://arxiv.org/abs/1608.07905

11. Appendix

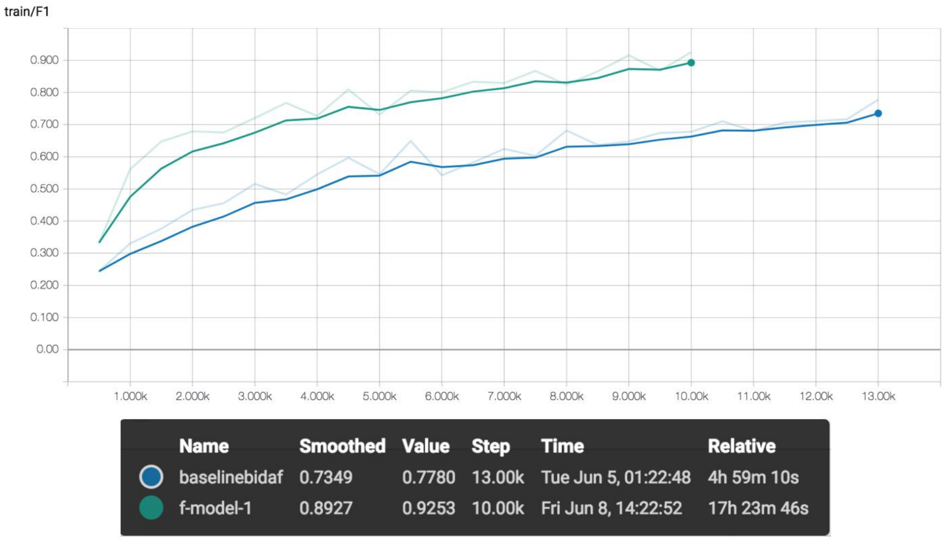
11.1. Train







	Name	Smoothed	Value	Step	Time	Relative
●	baselinebidaf	0.6297	0.6860	13.00k	Tue Jun 5, 01:22:48	4h 59m 10s
●	f-model-1	0.7969	0.8390	10.00k	Fri Jun 8, 14:22:52	17h 23m 46s



	Name	Smoothed	Value	Step	Time	Relative
○	baselinebidaf	0.7349	0.7780	13.00k	Tue Jun 5, 01:22:48	4h 59m 10s
●	f-model-1	0.8927	0.9253	10.00k	Fri Jun 8, 14:22:52	17h 23m 46s

11.2. Sampling

11.3. ”What”Types of Questions

”What”types of Questions corresponded to a wide variety of questions. The key takeaway is that “What” questions correspond to a diverse range of patterns and that presents a considerable challenge for the model. What questions, as noted above also draw from a range of different answer types.

Overall we saw 29 different samples that corresponded to What questions. Of the 29, our model presented the correct exact answer (i.e EM Score = true) for 14 of them, and incorrectly for 15 of them. The average F1 score was 0.553. This speaks to the extremes of the performance of the model: F1 scores were mostly either 1 or below 0.250, and scarcely in between.

We noticed that there were very few what questions that returned completely different answers to the ground truth. A lot of the answers that returned an EM = False, seemed to grasp the right answer, but also returned some extra words (which were not nonse and did add to the answer). A good example would be the following:

Context:
“in recent years ’the gate’ has opened in the city centre , a new indoor complex consisting of bars , upmarket clubs , restaurants and a 12-screen empire multiplex cinema . newcastle ’s gay scene the UNK pink triangle UNK is centred on the times square area near the centre for life and has a range of bars , cafés and clubs .”

Question: what is ’ the gate ’ ?
True answer: indoor complex
Predicted answer:
a new indoor complex consisting of bars , upmarket clubs , restaurants and a 12-screen empire multiplex cinema
F1 score: 0.250
EM score: False

The predicted answer returns a longer string than required. While the focus of the answer is right ("an indoor complex"), the answer provided is scored very low on the F1 score. This is of course because the F1 score suffers when accounting for precision and recall, with a lot of extra words not in the ground truth. This kind of data-set bias towards shorter answers would make sense for certain applications. It would also make sense if it was consistent. However, we found some answers that supported longer answers instead of shorter ones. An example of it is seen here:

Context:
the very high-speed backbone network service UNK came on line in april 1995 as part of a national science foundation UNK sponsored project to provide high-speed interconnection between UNK sponsored supercomputing centers and select access points in the united states . the network was engineered and operated by mci telecommunications under a cooperative agreement with the nsf . by 1998 , the UNK had grown to connect more than 100 universities and research and engineering institutions via 12 national points of presence with UNK (45 mbit/s) , UNK (155 mbit/s) , and UNK (622 mbit/s) links on an all UNK backbone , a substantial engineering feat for that time . the UNK installed one of the first ever production UNK (2.5 gbit/s) ip links in february 1999 and went on to upgrade the entire backbone to UNK .
Question: by 199 how many universities were connected
True answer:
by 1998 , the vbns had grown to connect more than 100 universities and research and engineering institutions via 12 national points of presence with ds-3
Predicted Answer: 100
F1 SCORE: 0.080
EM SCORE: False

Given this variety was found in a microcosm of the larger data-set, it is possible that there are lot of training examples that have similar contradictions. As a result the model cannot learn to favour smaller answers over longer ones (if this was the goal).

A lot of the what question errors came down to these kinds of 'off by a few word' problems. While the model was certainly able to pay attention to the right words in the context, it was not so good filtering at predicting the exact words in the ground truth and filtering for it. One possible fix is to condition the end probability on the start probability. The current model predicts the start location and the end location independently, given the final layer's activation: following the example of the Answer Pointer component of the 'Match-LSTM with Answer Pointer' model.

Further, we noted that lexical differences between the question words and the context words accounted for a large chunk of the errors. Often we found questions that included words that were not found in the context, though there words that corresponded to the same meaning. This is not surprising because according to Rajpurkar et al (2016) 9.1% of "Major correspondences between the question and the answer sentence require world knowledge to resolve."Let's consider an example of this:

Context:
in most jurisdictions (such as the united states) , pharmacists are regulated separately from physicians . these jurisdictions also usually specify that only pharmacists may supply scheduled pharmaceuticals to the public , and that pharmacists can not form business partnerships with physicians or give them ”kickback ”payments . however , the american medical association (ama) code of ethics provides that physicians may dispense drugs within their office practices as long as there is no patient exploitation and patients have the right to a written prescription that can be filled elsewhere . 7 to 10 percent of american physicians practices reportedly dispense drugs on their own .
Question: what are pharmacists forbidden to do ?
True Answer:
form business partnerships with physicians or give them ”kickback ”payments
Predicted answer: regulated separately from physicians
F1 SCORE : 0.143
EM SCORE: False

Here the word ’forbidden’ is not explicitly present and hence the attention and the model focuses on ”pharmacists are”as the start of the answer and probably the full stop as the end. However the target model would focus on ”pharmacists can not”as an indication to the start of the answer. This is likely because the model cannot equal forbidden to ”can not”. While the pretrained word embeddings should show a similarity between these phrases and hence influence the output- it doesn’t seem to be working here.

We found many more examples of how such difference in words caused the model to fail.

11.4. When, Where and Who

When, Where and Who questions generally performed really well. When questions had an average EM score of 90%; Where had 63%; Who had 70%.

We noticed that many of these types of questions were short and mostly involved syntactic variation. According to Rajpurkar et al (2016), syntactic variation accounted for the answers of around 64% of the data-set. For them, syntactic variation occurs when the ”syntactic dependency structure does not match that of the answer sentence even after local modifications.”Our model fairly well under these circumstances. This is likely because the glove word embeddings is able to identify the similarity between these syntactic variation.

Our model performed pretty well on questions where there were lexical variation too. An example is seen here:

Context:
the university of chicago is governed by a board of trustees . the board of trustees oversees the long-term development and plans of the university and manages fundraising efforts , and is composed of 50 members including the university president .
Question: who runs the university of chicago ?
True answer: a board of trustees
Predict answer: board of trustees
F1 SCORE: 1.000
EM SCORE: True

Here our model is able to identify that ’runs’ corresponds to ’governed’.

In general, we also found that because these questions are effectively looking for a specific type of answer (location, date, entity or person), the model was able to zone in on the right answer. For instance, when asked ’what cbs website provided a stream ?’, it was able to zone in on cbssports.com

Our model also quiet easily mastered the cases where the question is just the answer paraphrased into a declarative form (eg - question: ’when did jamaa islamiya renounce violence ?’, answer from context: ’renounced violence in 2003’).