
CharBiDAF with Self-Attention for SQuAD

Darrith Phan

Department of Computer Science
Stanford University
Stanford, CA 94309
darrithp@cs.stanford.edu

Zahra Abdullah

Department of Computer Science
Stanford University
Stanford, CA 94309
zahraab@cs.stanford.edu

Abstract

The Stanford Question Answering Dataset (SQuAD) is a reading comprehension dataset designed to challenge systems in “understanding” text. Given a paragraph and a question about the paragraph, the goal of the SQuAD challenge is to answer the question correctly. This paper explores the effectiveness of using a bi-directional attention flow (BiDAF) model with character-level embeddings and a self-attention mechanism on the SQuAD challenge. We additionally experimented with varying the hyperparameters of the model. On our best model, we were able to achieve an F1 Score: **64.938** and a EM Score: **61.302** on the test set in the non-PCE division.

1 Introduction

Question answering systems play a significant role in the field of Natural Language Processing. The SQuAD challenge reflects a system’s “understanding” of a given piece of text. As such, systems performing well on the SQuAD dataset could aid human understanding and retention of complex documents. The current best-performing models according to the SQuAD 2.0 leaderboard all use some form of pre-trained contextual embeddings, mostly from BERT [6]. Such methods have even surpassed human performance on the question-answering task. We opted to forego the PCE and instead experimented with character-level word embeddings and self-attention layers. These are included within a BiDAF model. We found that the best performance was achieved by a BiDAF model with character-level word embeddings while the full model (BiDAF + character-level embeddings + self-attention) came in a close second. The results (with various hyperparameter adjustments) are shown below.

2 Related Work

There exists a variety of research relating to text comprehension and many papers that specifically work on the SQuAD dataset. There are a few notable papers that we have taken inspiration from for the construction of our own model. The first study uses a Bidirectional Attention Flow network for machine comprehension, which is also the baseline model used for this project [2]. It also outlines an implementation of character-level word embeddings. Another study attempted to tackle reading and question answering by creating an architecture that does not rely on recurrent networks, but only convolution and self-attention [4]. The study’s construction and use of encoder blocks inspired us to develop a self-attention block for our model that’s similar to their implementation. While there are many more papers that discuss SQuAD, the two papers described above contributed the most to our own model.

3 Approach

3.1 Baseline

For our baseline model, we used a bi-directional attention flow model based on [2].

3.2 Architecture

We extended the baseline model by including character-level word embeddings [2] and self-attention [4]. Our model contains the following layers:

Embedding Layer: Let $w_1, \dots, w_{k_1} \in \mathbb{N}$ be input word indices, and let $c_1, \dots, c_{k_2} \in \mathbb{N}$ be character indices for the same input. We first perform an embedding lookup to convert the indices into word embeddings and character embeddings respectively. This is done for both the context and the question, producing word embeddings $cw_1, \dots, cw_{N_1} \in \mathbb{R}^{D_1}$ and $qw_1, \dots, qw_{M_1} \in \mathbb{R}^{D_1}$ for the context and question words, and producing character embeddings $cc_1, \dots, cc_{N_2} \in \mathbb{R}^{D_2}$ and $qc_1, \dots, qc_{M_2} \in \mathbb{R}^{D_2}$ for the context and question characters. We project each word embedding to an $\frac{H}{2}$ -dimensional vector using a linear layer without a bias term. We reshape the character embeddings and pass them through a single convolutional layer with a 5×5 kernel so that the output for the character embeddings for a given word is a vector of the same dimension as the corresponding projected word embedding. We then concatenate the corresponding projected word embedding and convolved character embedding to give a vector $h_i \in \mathbb{R}^H$. h_i is then passed through a Highway Network.

Encoder Layer: This layer uses a bi-directional LSTM to incorporate temporal dependencies between timesteps of the embedding layer’s output [1]. We also experimented with using a GRU in place of the LSTM.

Self-Attention Block: Our implementation is based on the QANet encoder block (cite QANet). We follow the QANet’s guidelines for the number of convolutional layers and output channels to use. We do not use the position encoding layer mentioned in the paper.

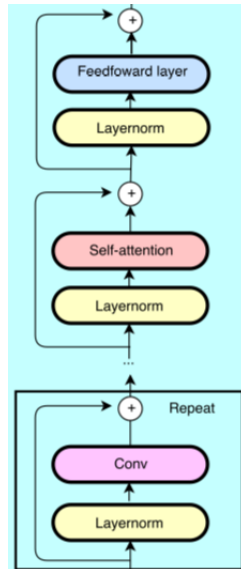


Figure 1: Self-attention block (based on QANet encoder block [4])

For the self-attention layer, we refer to [7]. Our model uses multi-head attention with 8 heads, with each head using scaled dot-product attention as described in [7]. The input and output of the self-attention block are both of dimension $2H$. The model includes a single self-attention block after

the encoder layer. Both the context and the query encodings are passed through this block.

Context-Query Attention Layer: This layer models a bi-directional flow of attention, from the context to the question, and from the question to the context [1].

Modeling Layer: This layer refines the sequence of vectors after the attention layer [1].

Self-Attention Block x 3: We use 3 self-attention blocks (sharing their parameters) in sequence. Each block has 2 convolutional layers.

Output Layer: The output layer produces a vector of probabilities corresponding to each position in the context [1].

The Highway Network, Encoder layer, Context-Query Attention layer, Modeling layer, and Output layer mentioned above are all implemented according to the description given in the default project handout for each of these components [1].

4 Experiments

4.1 Data

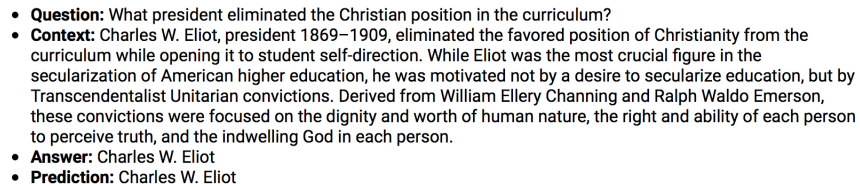
- 
- **Question:** What president eliminated the Christian position in the curriculum?
 - **Context:** Charles W. Eliot, president 1869–1909, eliminated the favored position of Christianity from the curriculum while opening it to student self-direction. While Eliot was the most crucial figure in the secularization of American higher education, he was motivated not by a desire to secularize education, but by Transcendentalist Unitarian convictions. Derived from William Ellery Channing and Ralph Waldo Emerson, these convictions were focused on the dignity and worth of human nature, the right and ability of each person to perceive truth, and the indwelling God in each person.
 - **Answer:** Charles W. Eliot
 - **Prediction:** Charles W. Eliot

Figure 2: Example paragraph and question with prediction for SQuAD.

Given a paragraph and a question about the paragraph, our model attempts to answer the question as correctly as possible. In order to train our model, we are using the provided SQuAD 2.0 machine comprehension dataset, which includes 129,941 examples.

4.2 Evaluation Method

We are currently using the quantitative metrics proposed in the original SQuAD challenge: Exact Match Score and F1 Score. An overview of each metric is defined below:

- **Exact Match Score (EM):** A binary measure on whether or not the model outputs exactly what is stated in ground truth.
- **F1 Score:** The harmonic mean of precision and recall: $F1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$

Moreover, we can qualitatively analyze our results by inspecting the text output our model predicts from a given paragraph and a question about that paragraph.

4.3 Experimental Details

In addition to running the baseline model, we ran two different models. The first model included only character-level embeddings, and the second model additionally included our QANet Encoder blocks. For standard training, we set the number of epochs to $num_epochs = 30$, a dropout probability of $dropout = 0.2$, a batch size of $batch_size = 64$, and used the Adadelata optimizer. With both of these models, we also experimented with various hyperparameters, listed below:

- **Learning Rates:** We decided to vary the learning rate with the following rates: 0.1, 0.3, 0.5, 0.7, 0.9, 0.99. We also experimented with a decaying learning rate.
- **Drop Out Probabilities:** We primarily used a dropout probability of 0.2, but we also experimented with 0.1 and 0.3.
- **Optimizers:** We primarily used Adadelata optimizer for our training, but when we were experimenting with the QANet encoder block we used Adam optimizer.
- **RNNs:** We primarily used the Bidirectional LSTM, but also replaced it with GRU for some trials.

4.4 Results

4.4.1 Character-Level Embeddings

Model	LR	Dropout	RNN	Dev NLL	F1	EM	AvNA
Baseline	0.5	0.2	LSTM	3.19	59.77	56.21	66.78
CharCNN	0.1	0.2	LSTM	2.84	61.59	58.36	67.92
—	0.3	0.2	LSTM	2.82	63.61	60.44	69.79
—	0.5	0.2	LSTM	2.96	65.37	61.97	71.99
—	0.7	0.2	LSTM	2.81	66.11	63.03	72.27
—	0.5	0.2	GRU	2.78	66.09	62.76	72.32
—	0.7	0.2	GRU	2.74	65.55	62.11	71.82
—	0.3	0.2	GRU	3.09	63.93	60.21	70.66
—	0.7	0.3	LSTM	2.75	65.27	61.92	71.43
—	0.9	0.3	LSTM	2.60	66.45	63.15	72.49
Test (Non-PCE)	0.9	0.3	LSTM	—	64.802	61.64	—
Test (Non-PCE)	0.9	0.3	LSTM	—	64.938	61.302	—

Figure 3: Results from training various models with the character-level embeddings on Devset, with the final row being the result from the Testset in the non-PCE division.

Overall, our model that used character-level embeddings was a significant improvement to the baseline model with respect to the EM score and the F1 score (Figure 3). The model performed better than expected, possibly because they provided much greater granularity to the embeddings. We also noticed that tweaking the learning rate for this model had mixed results on the overall performance, which was unexpected. First, we tried decreasing the learning rate, which resulted in similar or worse results to the baseline. However, increasing the learning rate did somewhat reduce overfitting to the training set, possibly because it prevented learning from truly converging. In addition to modifying the learning rates, we also noticed that replacing the bidirectional LSTM with GRU for the RNN had worse results and minimal effect on the overall running time. Dev plots for the models are shown in Figure 4.

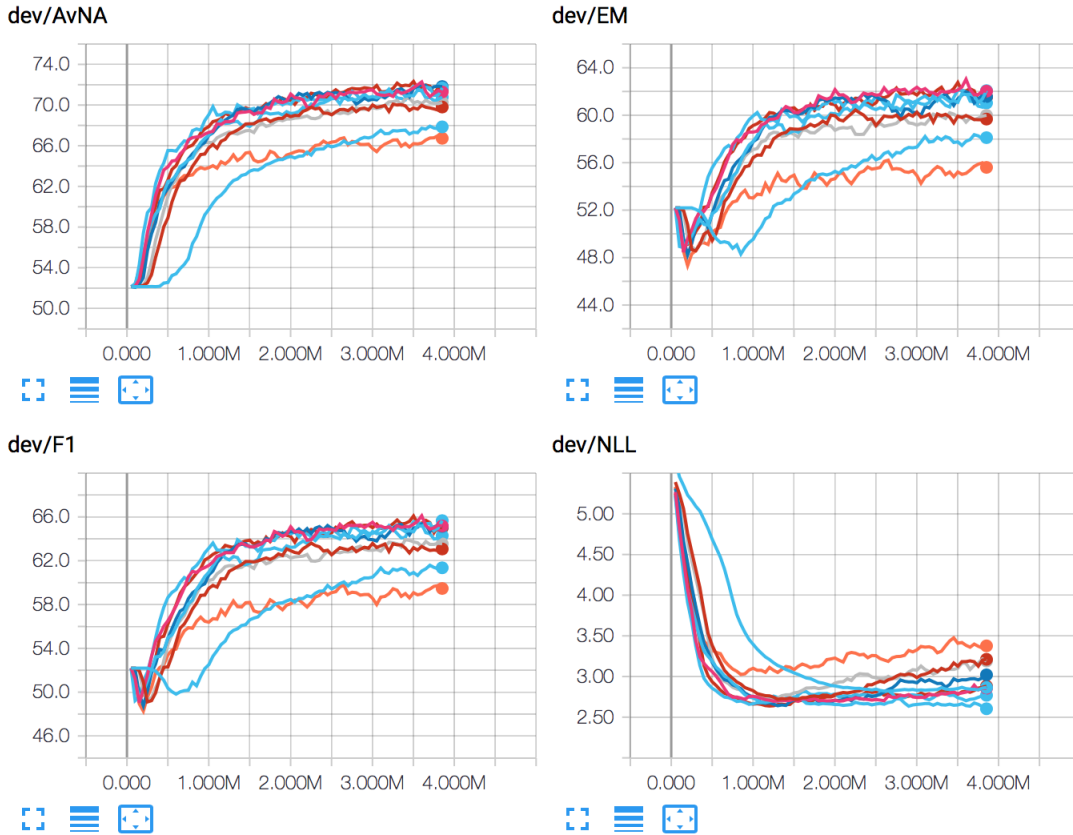


Figure 4: Plots of quantitative metrics for different models with character-level embeddings. The baseline model is in orange.

4.4.2 Character-Level Embeddings and Self-Attention Mechanism

Model	LR	Dropout	RNN	Dev NLL	F1	EM	AvNA
Baseline	0.5	0.2	LSTM	3.19	59.77	56.21	66.78
CharCNN + Encoder Block	0.5	0.2	LSTM	3.41	60.57	56.98	68.07
—	0.7	0.2	LSTM	3.21	61.40	57.62	68.59
—	0.99	0.1	LSTM	2.92	66.27	62.16	73.45
— (Adam Opt.)	0.0001	0.1	LSTM	5.16	51.89	51.89	52.26
Test (Non-PCE)	0.99	0.1	LSTM	—	63.167	58.833	—

Figure 5: Results from training various models with the character-level embeddings on Devset.

Overall, we see that the model that includes both character-level embeddings and the self-attention mechanism improved upon the baseline model with respect to EM and F1 scores (Figure 5). However, this model still performed worse than the one with only the character-level embeddings. We performed additional experiments by increasing the learning rate and received positive results, which was unexpected. One possible reason for the positive results at high learning rates is that the learning rate helped the model avoid local minima early on in training. Moreover, the plots suggest that these models could have benefited from additional training time, since they seem to have not reached a plateau in training yet. The Dev plots for the models are shown in Figure 6.

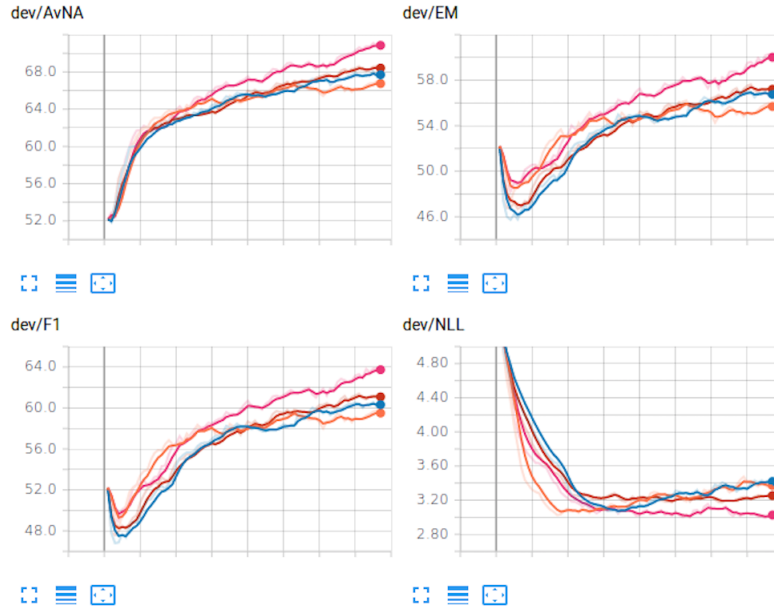


Figure 6: Plots of quantitative metrics for different models with character-level embeddings and the self-attention mechanism. The baseline model is in orange.

5 Analysis

- **Question:** Why is Warsaw's flora very rich in species?
- **Context:** The flora of the city may be considered very rich in species. The species richness is mainly due to the location of Warsaw within the border region of several big floral regions comprising substantial proportions of close-to-wilderness areas (natural forests, wetlands along the Vistula) as well as arable land, meadows and forests. Bielany Forest, located within the borders of Warsaw, is the remaining part of the Masovian Primeval Forest. Bielany Forest nature reserve is connected with Kampinos Forest. It is home to rich fauna and flora. Within the forest there are three cycling and walking trails. Other big forest area is Kabaty Forest by the southern city border. Warsaw has also two botanic gardens: by the Łazienki park (a didactic-research unit of the University of Warsaw) as well as by the Park of Culture and Rest in Powsin (a unit of the Polish Academy of Science).
- **Answer:** location of Warsaw
- **Prediction:** the location of Warsaw within the border region of several big floral regions

Figure 7: Baseline and CharCNN both presented the correct output as above, while the model with self-attention had no response (N/A).

Both our models performed better than the baseline model. In addition to the previous performance metrics, we see that both models also outperform than the baseline model on the Answer vs. No Answer (AvNA) metric. Since our models included character-level embeddings, they had much more success when answering questions about quantity or certain percentages. In terms of overall question answering, the baseline model and the model with only character-level embeddings were closer in behavior to each other than to the model with self-attention. Implementing self-attention allows a model to consider longer-range relationships between parts of the context and query. As a result, there's a chance it may overlook the proper answer to the question (see Figure 7). However, we clearly see the advantage of self-attention in Figure 8, where the model recognizes that the paragraph mentions two structures designed, a building and a garden. Our first two models incorrectly choose the first instance, which results in the wrong prediction for the architects. But, we can see that the self-attention model correctly identifies the proper architect of the garden.

- **Question:** Who designed the garden for the University Library?
- **Context:** Another important library – the University Library, founded in 1816, is home to over two million items. The building was designed by architects Marek Budzyński and Zbigniew Badowski and opened on 15 December 1999. It is surrounded by green. The University Library garden, designed by Irena Bajerska, was opened on 12 June 2002. It is one of the largest and most beautiful roof gardens in Europe with an area of more than 10,000 m² (107,639.10 sq ft), and plants covering 5,111 m² (55,014.35 sq ft). As the university garden it is open to the public every day.
- **Answer:** Irena Bajerska
- **Prediction:** Marek Budzyński and Zbigniew Badowski

Figure 8: Baseline and CharCNN both presented the incorrect output as above, while the model with self-attention had the correct response.

6 Conclusion

Overall, we find that incorporating character-level embeddings in the model provides a large improvement in text comprehension from the original baseline model. Similarly, the self-attention model showed clear progress over the baseline model as well. On the tests set in the non-PCE division, our first model with character-level embeddings was able to achieve an F1 Score: **64.938** and a EM Score: **61.302**. Our next model with self-attention was able to achieve an F1 Score: **63.167** and a EM Score: **58.833**. While our tests show that the model including the self-attention mechanism does not perform better than the model with only character-level embeddings, we believe that modifications to the hyperparameter space in the future could lead to more promising results.

7 References

- [1] (Original Handout) CS 224N Default Final Project: Question Answering on SQuAD 2.0
- [2] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.
- [3] R-Net: Machine Reading Comprehension with Self-Matching Networks <https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf>
- [4] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. Qanet: Combining local convolution with global self-attention for reading comprehension. *arXiv preprint arXiv:1804.09541*, 2018.
- [5] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *arXiv preprint arXiv:1608.07905*, 2016.
- [6] SQuAD2.0 The Stanford Question Answering Dataset <https://rajpurkar.github.io/SQuAD-explorer/>
- [7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. *arXiv preprint arXiv:1706.03762*, 2017.