Self-Attention is All You Need

Brian Hill *bwhill@stanford.edu* & Joanne Zhou *joannezhou@stanford.edu*

Background

Many people use search engines for information retrieval, and it is important to return the most relevant results. The SQuAD dataset aims to test models on their ability to return information from a narrower scope, namely the correct answer to a question from a paragraph-long passage. SQuAD 2.0 includes unanswerable questions. Our model improves upon the BiDAF baseline by adding in character embeddings, answer pointer network, ensembling, and the best performing of three self-attention methods.

Example Passage, Question, and Answer

Passage: Beyoncé Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".

Question: In what city and state did Beyoncé grow up?
Answer: Houston, TX

Question: What is Beyoncé's most recent album?
Answer: N/A (Passage cannot answer question)

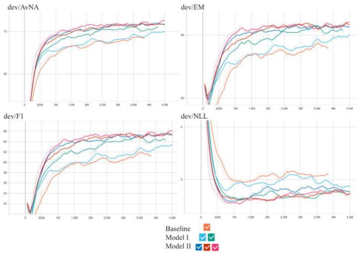
Experiments & Analysis

LSTM vs. GRU:
Both performed similarly in terms of performance, but GRU was slightly faster so we used it in all our RNN layers.

Self-Attention:
The model performance was highly sensitive to the type of self-attention used. We implemented R-Net's [2] gated self-attention in both additive and dot product forms, but observed minimal change from the baseline model results, leading us to believe the first bi-directional attention was sufficient. With the advice of our mentor Vincent Li, we then implemented residual self-attention modeled in [3], with great success. Our main takeaways from this experimentation is that:

1) An additional layer of self-attention after the BiDAF attention may not confer additional value, as the context-question alignment has already been performed.
2) A larger hyperparameter search may be required to find optimal hyperparameters specific to the added self-attention layer.

Ensemble:
We decided to ensemble Model I and Model II to improve performance, and chose the model with the best combined EM performance since our training script chose the best model by F1 performance, and we wanted to consider both metrics.



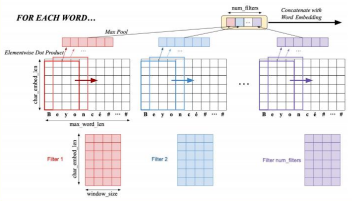
Dev Set Results		
Model	**EM (Exact Match)**	**F1 (Approximate Match)**
Baseline BiDAF	58.965	62.281
(I) Character Embeddings + Answer Pointer Network	61.586	65.143
(II) Character Embeddings + Residual Self Attention	62.729	66.201
Ensemble: 2 (I) & 3 (II)	**65.233**	68.409
Ensemble: 0 (I) & 3 (II)	65.115	**68.514**
Ensemble: 1 (I) & 3 (II)	65.199	68.489

Conclusions

1) Models I and II were both successful in performing above the baseline, and ensembling them performed even better, with over 65% exact match. Further, we observed better performance on the test set than the development set, indicating that our results are extendable to new data.
2) Based on our self-attention experiments, adding additional complexity to a model will not always improve results unless the complexity relieves a specific bottleneck.
3) Residuals are very important in preserving information between attention layers, in this way our model incrementally approached a more transformer-like architecture.

Test Set Results	
Model	Ensemble: 2 (I) & 3 (II)
EM	65.934
FL	68.909

Methods

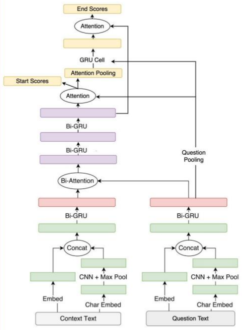


Character-Derived Word Embedding

In addition to the pretrained GloVE word embeddings, we wanted to pick up on meaning in sub-words to handle unseen or composite words. We implement a CNN that scans over the randomly initialized character embeddings for each word using hidden size number of filters to create a character-derived word embedding that is concatenated with the GloVE word embedding before passing into the encoding layer. We employ a window size of 4 to capture most prefixes (a hyperparameter search could be easily performed with other sizes in future work). In our experiments the character embedding had a large improvement on model performance, as it was effectively able to pick up on sub-word components.

Model I: BiDAF + Pointer Networks

Starting from the baseline BiDAF model, we add character embedding and use the pointer networks layer implemented by the R-Net paper [2] to predict the start and end probability of the answer. The initial hidden state for the pointer layer is an attention-pooling of the question encodings. An attention mechanism is used to predict the start and end position probabilities.



Pointer network layer:

1. Attention pooling of question encodings

$$s_j = v^T tanh(W_u u_j + W_V V^Q)$$

$$a_i = exp(s_i) / \sum_{j=1}^{q\_len} exp(s_j)$$

$$r^Q = \sum_{i=1}^{q\_len} a_i u_i$$

2. Get start probability

$$s_j^{start} = v^T tanh(W_v v_j + W_r r^Q)$$

$$a_i^{start} = exp(s_i^{start}) / \sum_{j=1}^{c\_len} exp(s_j^{start})$$

3. Set up pointer network for end probability

$$c^{start} = \sum_{i=1}^{c\_len} a_i^{start} v_j$$

$$h^{start} = GRUcell(c^{start}, r^Q)$$

4. Get end probability

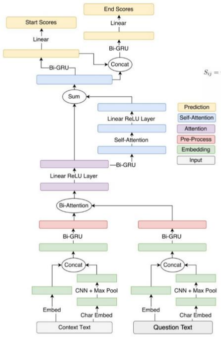
$$s_j^{end} = v^T tanh(W_v v_j + W_h h^{start})$$

$$a_i^{end} = exp(s_i^{end}) / \sum_{j=1}^{c\_len} exp(s_j^{end})$$

where u_j is question encoding after the encoding layer and v_j is context encoding after the last modeling layer

Model II: BiDAF + Residual Self-attention

Following the model structure proposed by Clark el. al. [3], we implement a passage-to-passage self-attention layer, and modify the output layer. The self-attention output is additionally summed with the BiDAF attention output.



Residual self-attention:
Output from the BIDAF attention is passed through a linear ReLU layer, and a Bi-GRU layer, obtain passage representations *c*

$$S_{ij} = w_{sim}^T [c_i; c_j; c_i \odot c_j] \qquad \bar{S}_{i,:} = softmax(S_{i,:}) \in \mathbb{R}^N \ \forall i \in \{1, ..., N\}$$

$$a_i = \sum_{j=1}^{N} \bar{S}_{ij} c_j \in \mathbb{R}^{2H} \ \forall i \in \{1, ..., N\}$$

Diagonal of similarity matrix is set to *-inf* before taking the softmax.

Attention output is passed through another linear ReLU layer and added to the previous BIDAF attention output.

Output layer:
Sum of attention is passed to a Bidirectional GRU layer and a linear layer to get the start position probability. The BiGRU layer output is concatenated with the attention sum, and passed to another BiGRU + linear layer to get the end position probability.

Ensembling

We train model I with two different random seeds and model II with three different random seeds. When predicting on a test/dev split, we average the output probabilities across the combination of models. Different combinations of the models are tested.

Implementation Details

hidden size = 100; dropout probability = 0.2; epoch num = 35; batch size = 64
word vector = GloVe 300 dimensional; character embedding dimension = 100
optimizer: Adadelta; loss: sum of the negative log-likelihood (cross-entropy) loss

References & Acknowledgments

[1] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hajishirzi Hananneh. Bi-directional attention flow for machine comprehension. In *ICLR 2017 Confrence Paper*, 2017.
[2] Natural Language Computing Group. R-net: Machine reading comprehension with self-matching networks. In *Association for Computational Linguistics (ACL)*, 2018.
[3] Christopher Clark and Matt Gardner. Simple and effective multi-paragraph reading comprehen- sion. *CoRR*, abs/1710.10723, 2017.

We want to thank our project mentors, Vincent Li and Kamil Ali, for their conversations with us at all stages of our model construction, as well as their helpful feedback on our proposal and milestone. Further, we want to thank Prof. Manning and Anne Golding for organizing the course and making this project possible.