Attention predicts "Nothing"

Samidh Pratap Singh [1] Mayank Gupta [1]

[1]Department of Computer Science, Stanford University

Introduction

In this project our goal is (1) to experiment with architectural improvements (embedding, attention or output layers) on BiDAF [1] baseline architecture provided to us with an eye on the performance on SQuAD 2.0 dataset, (2) and to study the impact of different attention mechanisms on the ability to understand text. We present results of adding a character embedding layer, question-passage coattention, passage self-attention, multi head attention and dynamic iterative decoder to the baseline BiDAF model. Character embedding provides a large jump (6%, F1 Score) in performance whereas attention layers provide only a marginal improvement (1.5%, F1 Score) on the bi-directional attention present in the baseline. Our single best model, achieves F1 score (68.81) and EM score (65.05), and ensemble model achieves **F1 score (70.62) and EM score(67.94)** at dev set. This puts us at **5th place at dev & test leaderboards**.

Background

1. We started with the data & the default implementation provided to us for IID track for BiDaf model, we refer to this as baseline everywhere.
2. We took inspirations from architectures in papers such as DCN [2], R-Net [3], and transformer lecture for trying out different attention mechanisms.
3. Lot of related work has been done for decoder/output architectures as well, such as DCN [2] and match-LSTM.

Figure 1. Overview of Bidaf architecture

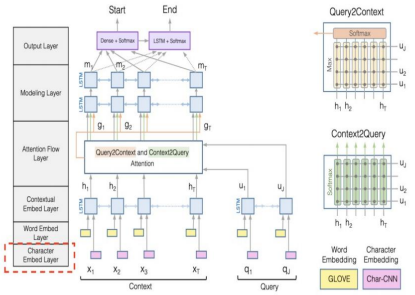

Figure 1: BiDirectional Attention Flow Model

Methods

1. **CharEmbeddings:** 100 Conv-1d filters each for kernel sizes 3 5. BatchNorm also helped in accuracy.
2. **CoAttention and Passage Self attention:** We implemented Co-attention from DCN paper and self-attention from R-net paper from scratch. Fusing Co-attention with Bidaf attention improved our scores.
3. **MultiHeadAttention:** We experimented with 4-headed and 8-headed attention. Improved performance by 1% over single headed attention.
4. **Iterative Decoder:** Implemented iterative decoder from DCN paper. In an experiment, we seeded the iterative decoder with Bidaf decoder output, but it didn't help with the performance.
5. **Hyper Parameter Tuning:** We experimented with exponential decay in learning rate, and also different values of dropout probabilities: {0.15. 0.2. 0.3}.
6. **Ensembling:** We combined our 10 best models and tried weighted average and majority voting ensembling and achieved 2% gain from it.
7. **Partial loading of parameters** from a previously trained model enabled us iterate faster over hypothesis and enabled transfer learning. ☺

Experiments

We performed experiments on top of the provided baseline code of BiDaf model. we implemented all methods explained in previous section from scratch in Embedding, attention and output/decoder layers and achieved following results:

Model	F1	EM	AvNA	Dev Loss
Baseline	61.29	57.84	68.01	3.08
+ char-emb	67.37	64.22	72.88	2.59
+ coattention	61.89	58.54	68.53	3.02
+ self-attention	61.49	58.36	68.19	3.04
+ co- & self-attention	62.84	59.27	69.53	2.98
+ char-emb + coattention	66.81	63.27	72.90	2.67
+ char-emb + co- & self-attention	68.23	65.17	74.09	2.51
+ iterative-decoder	66.44	63.59	72.24	2.58
multihead-attention	**68.81**	**65.05**	**74.69**	**2.57**
Ensemble (majority-voting)	70.51	**67.97**	74.95	2.63
Ensemble (weighted avg by F1)	**70.62**	67.94	**75.08**	2.63
Ensemble (test leaderboard)	69.208	66.627	NA	NA

Table 1. Results of Experiments

Analysis

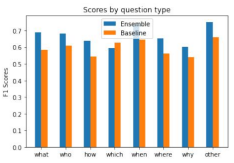

Figure 2. Analysis by question type

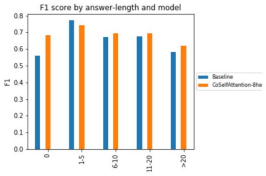

Figure 3. Analysis by answer length

Conclusions

1. Our model improves the prediction of "No answer" by 13% in F1 score!
2. The primary driver of improvement is Char embedding layer. BatchNorm helps stabilize the training and improves performance.
3. Co-Attention and Self-Attention individually performed poorly, but fusing them together and using multi head attention improved the performance.
4. Iterative Decoder approach in DCN didn't improve performance on its own. But using this model in the ensemble increased the performance. Further analysis and experiments can be performed to effectively utilize it.

References

[1] Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi.
Bidirectional attention flow for machine comprehension.
CoRR, abs/1611.01603, 2016.

[2] Caiming Xiong, Victor Zhong, and Richard Socher.
Dynamic coattention networks for question answering, 2018.

[3] Natural Language Computing Group.
R-net: Machine reading comprehension with self-matching networks.
May 2017.