Exploring Attention in Question Answering Models

Ethan Shen, Anav Sood

Motivation

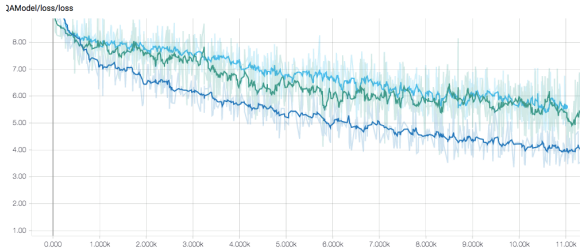
Machine Comprehension (MC) and Question Answering (QA) are complex problems that have become approachable through the lens of deep learning. Due to their high level of difficulty, MC and QA tasks are often used as a benchmark for measuring the progress of NLP.

Problem Statement and Dataset

Given a context and question, find a starting and end position in the context that correspond to an answer to the given question. To explore solving the problem we use the the SQuAD dataset.

Experiments with Attention

Using the framework of BiDAF, a model designed to contain a bi-attention layer, we implement bi-attention, co-attention, and self-attention layers in an effort to see how well attention layers translate to different frameworks.



Train loss of bi (dark blue), co (light blue), and self-attention (green) over eleven thousand iterations.

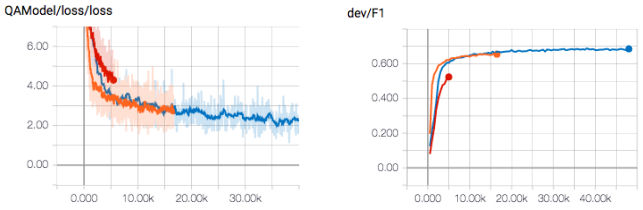
Co- and self- attention performed well, but not as well as in their original framework. With more training, they may have performed at par with bi-attention, but were generally more complex layers and took longer to train.

Future work could involve developing a model structure where all such layers perform at a high level, or using two or more such layers in the same model.

Hyper-parameter Tuning

Final chosen hyper parameters:

- Hidden state size – 200
- Optimizer – Adam with initial learning rate 0.001
- Regularization – Dropout with parameter 0.15
- BiRNNs and RNNs – LSTM



Train loss and dev F1 score for BiDAF model with baseline parameters (orange), baseline but with Ada-Delta optimizer (red), and baseline but with LSTMs (blue).

With these hyper-parameters, we add character level word embeddings with CNNs and change our start-stop position output to i, j = argmax p_1(i)p_2(j), where $i \leq j \leq i + 10$.

Results with BiDAF Ensemble

We ensemble 10 BiDAF models using a majority voting. The ensemble reaches a dev F1 and EM scores of 77.84 and 68.39.

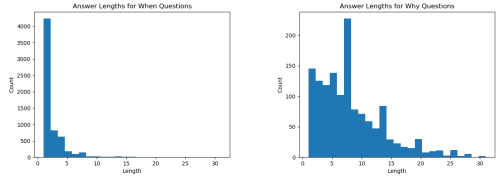
Table 1: Results of bi-attention models on dev set. Each subsequent model includes the modifications of the one before it.

Model	F1(%)	EM(%)
BiDAF (ensemble)	77.84	68.39
BiDAF (new span)	76.10	66.23
BiDAF (with CNN)	73.53	63.76
BiDAF (no CNN)	69.37	58.11
Baseline	44.22	35.07

Table 2: Performance of a single BiDAF (with CNN layer) on varying question types in the dev set.

Question Type	F1(%)	EM(%)
When	85.64	79.63
Who	78.03	71.02
Which	76.90	66.13
How	75.93	66.01
Where	73.75	63.43
What	74.52	61.81
Why	65.64	41.77

As shown by Table 2, the model performs poorly on "why" questions and very well on "when" questions. Shown below, the model struggles with questions with long answers. Likely such answers/questions are more ambiguous and complex.



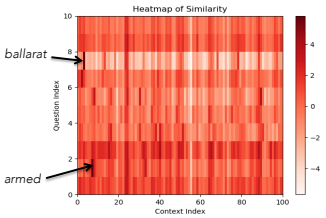
Distribution of answer lengths for "when" and "why" questions.

Error Analysis: Miss Attention

Context: *in 1854 at ballarat there was an armed rebellion against the government of victoria by miners protesting against mining taxes (the ``eureka stockade'') . this was crushed by british troops …*

Question: *what armed group stopped the uprising at ballarat ?*

True Answer, Predicted Answer: *british troops, miners*



Heatmap of similarity shows the model pays attention to words closer to miners rather than British Troops, partially because of high similarity encodings of words repeated in the question and context

Error Analysis: Syntactic Ambiguity

Context: *…chemical energy released in combustion . combustion hazards also apply to compounds of oxygen with a high oxidative potential , such as peroxides , chlorates , nitrates , perchlorates , and dichromates because they can donate oxygen to a fire .*

Question: *peroxides , nitrates and dichromates are examples of what type of compounds ?*

True Answer, Predicted Answer: *compounds of oxygen with a high oxidative potential, combustion hazards.*

Conclusion

When used with a proper model foundation, attention layers are very effective in QA and MC tasks. We find that character level embeddings, ensembling, and tweaking output spans are all effective in increasing scores. To make further progress it is important to get the model to truly understand the question rather than find suitable answers around particular similar/important words.