A Study of Attention in Deep Learning Models for Question Answering

William Locke
Stanford University
wlocke@stanford.edu

Abstract

A question answering system answers questions posed in natural language based on a passage of text. The accuracy of such systems is commonly scored by comparing the system's answers against human-generated answers. Such systems can be developed using deep learning models, provided a large enough dataset containing human-generated reading comprehension question/answer pairings. One such deep learning model architecture involves encoding the passage along with the question as word embeddings, feeding through an encoder-decoder layer with an accompanying attention mechanism. The experimental evaluations discussed herein focus on how varying the attention layer of such an architecture affects model performance when measured using the Stanford Question Answering Dataset (SQuAD).

1 Introduction

In assessing the requirements of a question answering system, a number of baseline requirements are immediately apparent. Firstly, given that the system will be required to answer any question posed in natural language about any given passage of text, the scope of possible passages and questions are any that can be generated using the natural language in question. Given that the questions may be posed using words not found in the training set, the system will need a way of interpreting such words. Essentially, every word in the chosen vocabulary will need to have a representation that captures its general pattern of usage such that questions posed with words not found in the training set can be accurately answered.

One solution is to use a set of word embeddings such as GloVe[1]. Trained on large corpora such as Wikipedia, word embeddings learn relationships between words whereby the general usage pattern of large vocabularies can be captured and encoded as vectors. While GloVe vectors capture the nature of words' relationship with other words, capturing the relationship between any given sequence of words with another requires additional layers of learning.

Since reading comprehension requires an understanding of language structure, a model designed to achieve question answering must be capable of capturing information about relationships among groups of words such as phrases. Furthermore, given the possibility of lengthy passages consisting of many sentences, the effects of such relationships must extend beyond the phrase level to the broader context of the passage. Recurrent neural networks provide one method for capturing such meaning. In the realm of sequence to sequence problems such as machine translation, models that employ recurrent neural networks to first encode an input sentence and a decoder network to produce the corresponding output sentence have proven effective[2]-[4].

A final layer that has proven critical to high performing question answering systems[5], along with sequence to sequence models in general[6], is called attention. In this context the attention layer act as a means to focus on question words that have a particular relationship with context words. Deep learning models with architectures such as this, complete with attention, have proven effective in developing question answering systems that perform highly when tested against datasets such as SQuAD[7].

2 Approach

Given the general architecture described above, the goal of this paper is to measure the performance gains achieved by several varieties of attention against a baseline implementation. In particular, this paper will look at attention implementations of the systems described in *Effective Approaches to Attention-based Neural Machine Translation*[8] and *Bi-Directional Attention Flow for Machine Comprehension*[9] referred to in this document as Luong and BiDAF attention respectively. While a baseline implementation of a question answer system using word embeddings, an encoder-decoder and attention yields promising results, there is nonetheless a considerable difference in scores obtained by high performing implementations. The goal of these experiments will be to analyze contributions in performance gains achieved by introducing only the attention mechanism while keeping all other aspects of the baseline model constant. Subsequent analysis will focus on gains in overall measures of performance as well as patterns in the passage-question pairs in which any gains are realized.

3 Model architecture

The process of developing a parent architecture to house the components of the models in question was aided by the modular nature of each along with the fact that the layers of each architecture are paralleled in the baseline. Below is a summary of each of the layers in the system along with descriptions the variants contributed by each system.

Table 1: Model architecture

	Baseline	**Luong Attention**	**BiDAF Attention**
Output layer	Softmax		
Modeling layer	Fully connected	2 layer LSTM	
Attention layer	Basic dot-product attention	Luong global attention	Attention flow layer
Encoding layer	Bidirectional GRU		
Embedding layer	Word embeddings		

3.1 Embedding layer

In all model variants the input passage $Q = \{w_t^Q\}_{t=1}^M$ along with question $P = \{w_t^P\}_{t=1}^N$ are converted to word-level embeddings $\{e_t^Q\}_{t=1}^M$, $\{e_t^P\}_{t=1}^N$.

All models use GloVe vectors for the word embeddings. Although various tweaks may be made to the embedding layer to improve model performance, in our case the word embedding part of this

layer will be held constant. 300-dimensional GloVe vectors trained on a 6 billion token corpus will be used across the board. Furthermore, these word vectors will not be trainable.

3.2 Encoding layer

In the encoding layer, embeddings of the question and passage are passed through a bidirectional recurrent neural network (RNN) resulting in a new representation of each. A bidirectional GRU is used in this layer, with the forward and backward output sequences concatenated.

$$u^Q = BiGRU_Q(e_1^Q, ..., e_M^Q)$$

$$u^P = BiGRU_P(e_1^P, ..., e_N^P)$$

While the BiDAF paper describes using LSTM cells in this layer, GRUs were found to perform similarly in testing and consequently were employed in the final implementation given their relative computational cheapness.

3.3 Attention layer

The baseline model employs basic dot-product attention, whereby the encoded passages attend to the encoded questions.

$$S^i = [u_i^{P^T} u_i^Q, ..., u_i^{P^T} u_M^Q] \in \mathbb{R}^M$$

$$\alpha^i = softmax(S^i) \in \mathbb{R}^M$$

The resulting attention distribution is then used to take a weighted sum of the encoded questions producing the attention output. This output is then concatenated to the encoded passage to give blended representations.

$$a_i = \sum_{j=1}^{M} \alpha_j^i u_j^Q \in \mathbb{R}^{2h}$$

$$b_i = [u_i^Q; a_i] \in \mathbb{R}^{4h} \qquad \forall_i \in \{1, ..., N\}$$

3.3.1 Luong attention

In the original paper from which this attention implementation is derived, *Effective Approaches to Attention-based Neural Machine Translation* describes two variants. These are referred to as "global attention" and "local attention." The implementation used in this case will be global attention. The implementation mirrors that of the baseline attention implementation, however the "score function', symbolized as S in the baseline equations, uses the "general" form whereby W_a is a trainable weight matrix.

$$S^i = [u_i^{P^T} W_a u_i^Q, ..., u_i^{P^T} W_a u_M^Q] \in \mathbb{R}^M$$

3.3.2 BiDAF attention

The attention flow layer of the BiDAF model involves computing attention in both directions. In the first instance, a similarity matrix calculated between the encoded passage and question is calculated where α is a trainable scalar function encoding the similarity between the two inputs. A shared similarity matrix $S \in \mathbb{R}^{N \times M}$ is calculated between the passage and question embeddings such that S_{nm} indicates the similarity between the n-th passage word and the m-th

question word, where function α is defined below with $w_{(S)} \in \mathbb{R}^{6h}$ being a trainable weight vector.

$$\alpha(u^P_{:n}, u^Q_{:m}) = w^T_{(S)}[u^P_{:n}; u^Q_{:m}; u^P_{:n} \odot u^Q_{:m}]$$
$$S_{nm} = \alpha(u^P_{:n}, u^Q_{:m}) \in \mathbb{R}$$

Passage-to-question attention is then calculated as follows

$$\widetilde{u^Q_{:n}} = \sum_m softmax(S_{n:})_m u^Q_{:m}$$

Followed by question-to-passage attention

$$\widetilde{u^P_{:m}} = \sum_n softmax(max_{col}(S))_n u^P_{:n}$$

Our attention layer output is then computed by combining passage embeddings and attention vectors as follows

$$b = [u^P; \widetilde{u^Q}; u^P \odot \widetilde{u^Q}; u^P \odot \widetilde{u^P}] \in \mathbb{R}^{8h \times N}$$

3.4 Modeling layer

For the baseline model, the output of the attention layer b is fed through a fully connected layer with a ReLU non-linearity to produce the modeling layer output l.

$$l_i = ReLU(W_{FC} b_i + v_{FC}) \in \mathbb{R}^{h \times N}$$

3.4.1 BiDAF modeling layer

Experiments that employed the BiDAF attention layer were always coupled with the following modeling layer. The output of the attention layer is fed through two layers of a bidirectional LSTM resulting in a matrix $l \in \mathbb{R}^{2h \times N}$

$$l = BiLSTM_{2-layer}(b) \in \mathbb{R}^{2h \times N}$$

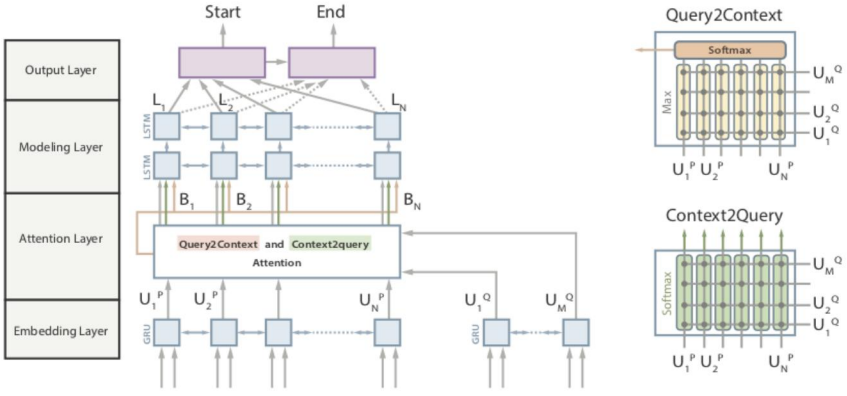

Figure 1: Baseline with BiDAF attention and modeling layers[8]

3.5 Output layer

The logits representing the index of the answer start word in the passage and the end word index, are by passing the output of the modeling layer l through separate fully connected layers with a ReLU activation function. The corresponding probability distributions for start and end positions are generated by passing the end and start logits through a softmax function.

$$logits_i^{start} = ReLU(W_{start} l_i + v_{start}) \quad \forall_i \in \{1, ..., n\}$$

$$p^{start} = softmax(logits^{start}) \in \mathbb{R}^n$$

($logits^{end}$ and p^{end} are computed equivalently)

4 Training details

The following configurations were run for fifteen thousand iterations each.

- *Baseline*
- *Baseline with Luong attention*
- *Baseline with Luong attention and BiDAF modeling*
- *Baseline with BiDAF attention and modeling layer*

Other than these layer configuration differences, all other factors were held constant. 300-dimensional GloVe vectors trained on a 6 billion token corpus were used. Adam optimizer was used with a learning rate of 0.001. Dropout was applied to all RNN cells at a rate of 0.2. Padding and clipping of questions and passages was implemented to achieve uniform length inputs. The maximum question and passage lengths were set to 30 and 600 respectively in all training runs reported. Out of vocabulary words–those not found among the word vector set–were handled by setting to zeros. A hidden layer size of 100 and a minibatch size of 40 was used in all cases.

5 Results

All model variants were run for a minimum of fifteen thousand training iterations (6 epochs) each. All models were run on an NVIDIA Tesla M60 GPU. The baseline model trained the fastest approximately 0.9 seconds per iteration (3.75 hours to train), followed by the Luong Attention model at 3.5 seconds (15.5 hours), then the BiDAF attention model at 3.9 seconds per iteration (16 hours to train), then the model with Luong attention and the BiDAF modeling layer at 5.5 seconds per iteration (23 hours to train). Figure 1 shows the F1 performance of the variants compared against that of the baseline. The baseline model scored a peak dev set F1 score of around 0.39 by the end of 6 epochs, while the addition of Luong attention resulted in a peak score of 0.61. Adding the BiDAF modeling layer bumped the peak score to 0.65, while the model with BiDAF attention and modeling scored the most highly at 0.71. The corresponding exact match scores reached 0.29, 0.46, 0.49 and 0.55 respectively.

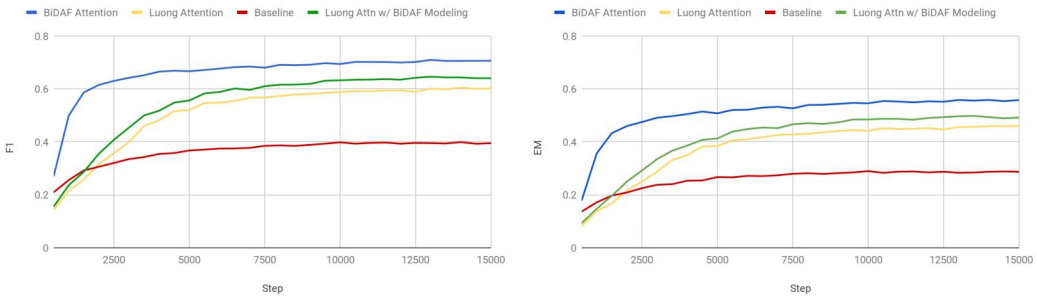

Figure 2: F1 and EM performance comparison

5.1 Error analysis

The accuracy gains realized by the attention mechanisms introduced can be studied in terms of the questions consistently answered incorrectly by the baseline model but answered correctly by the modified models. In doing this analysis questions were grouped into six categories with the possibility of category overlap.

- *Unknown question words* were defined as questions where greater than 10% of the tokens in the question were unknown.
- *Start, end mismatches* were defined as any prediction where the start token came after the predicted end token or where the end token was predicted to be more than 20 words in front of the start token. *Long answers*
- *Long answer* questions were those with answers with four or more words
- *Long passage* questions involved passages with 150 words or more
- *Short passage* questions had passages under 150 words
- *Imprecise boundary* questions are those predicted partially correctly

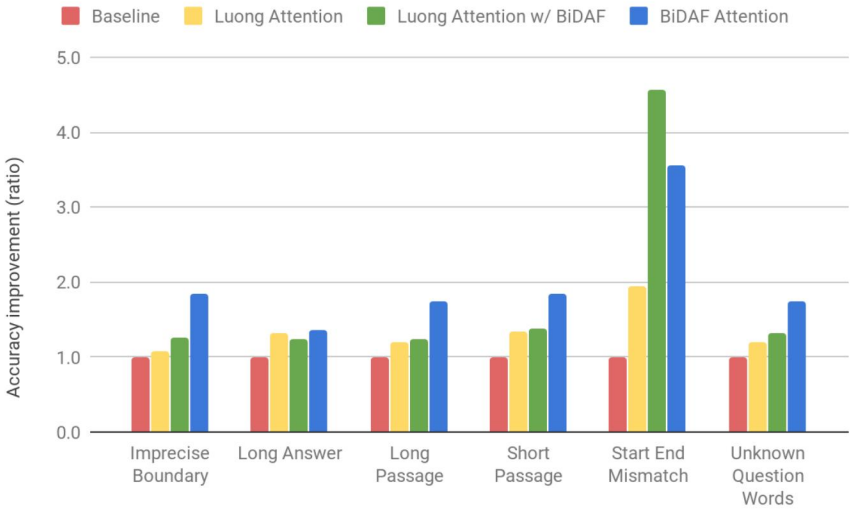

Figure 3: EM gains by question category relative to baseline

The results of this analysis are displayed in figure 3 where the relative occurrence of each question category is normalized to display only the ratio of improvement of the variant models against the baseline. While all categories show improvement, there appear to be modest gains in question with *long answers* and a greater than average reduction in the occurrence of *start, end mismatches*.

5.1.1 Long Answers

Investigating the long answer question category further shows below average exact match scoring with the BiDAF model only achieving 0.45 versus an average of 0.55. Examining individual question-answer pairings yields examples such as displayed in table 2 where seemingly correct answers are scoring poorly as the result of verbose human answers. Further analysis revealed that the ratio of partially correct answers to exact matches in this question category to be one and half times the average indicating that the difficulty in improving the success in this category could be in part due to verbose human answers rather than a model limitation specific to answer length.

Table 2: Example long answer question

Context	… Newton was limited by Denver's defense, which sacked him seven times and forced him into three turnovers, including a fumble which they recovered for a touchdown…
Question	Who was limited by Denver's defense?
Answer	Newton was limited by Denver's defense
Prediction	Newton
Models	All models

5.1.2 Start, end mismatch

Interestingly, a large degree of the gains in F1 and EM accuracy achieved by the improved models were attributable to improvements in this area. Such errors account for around 30% of the baseline errors, while only 15% of the model with BiDAF attention (with an absolute improvement of almost four times). Table 3 shows an example of a question answered incorrectly by the baseline model but correctly by the variants. Given this improvement occurs without a specific mechanism that feeds information about the start token prediction to the end token predictor, it is worth thinking about the mathematical implications of improving the accuracy of two independent predictions. For instance, improving the single token prediction accuracy from 0.5 to 0.75 (an improvement ratio of 1:1.5) would result in a ratio of improvement of 1:2.25 (the square of the previous ratio). As such, it may be the case that single token prediction improvement among this group of previously incorrectly answered questions goes a long way to explaining the outsized degree of improvement.

Table 3: Example start, end mismatch question

Context	… encompasses six libraries that contain a total of 9.8 **million** volumes … the John Crerar Library contains *more* than 1.3 million volumes in the biological , medical and physical sciences and collections in general science and the philosophy and history of science …
Question	How many volumes does the John Crerar Library roughly hold?
Answer	more than 1.3 million
Prediction	<null> start: “more”, end: “million” (first occurrence)
Models	Incorrect: baseline, correct: all variants

5.1.3 Answer overlap

A final piece of error analysis involves looking at the overlap between questions answered

correctly by the baseline and those answered correctly by the variants. In figure 4, questions answered correctly by baseline but incorrectly by the variants are represented in the upper-middle section labeled *degrades baseline*. While the ratio of such questions is low, it is far from the case that all questions answered correctly by the baseline were correctly answered the variants. Such a result reiterates the non-deterministic nature of such models along with the fact that changes that improving the overall accuracy are not necessarily done so by incrementally reducing the set of questions answered incorrectly.

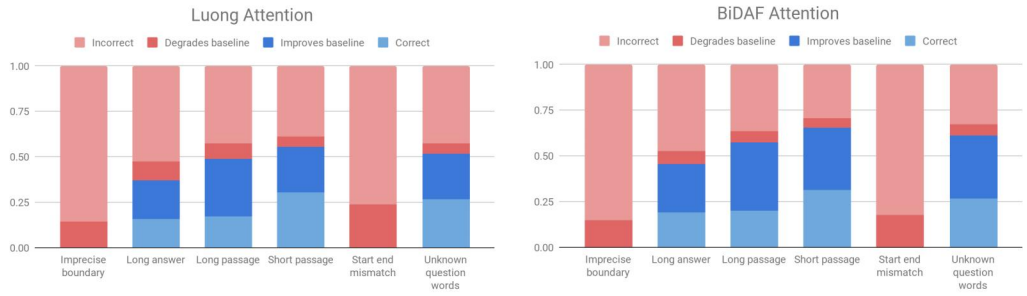

Figure 4: improvement vs. degradation of baseline by question category

6 Conclusion

Focusing on the attention layer while keeping all other aspects of the model implementation constant proved useful in deepening my understanding the importance of this layer to the overall architecture. While a mechanism that can learn a focused representation of an input and output sequence pairing on a per input word basis is both intuitive and effective in practice, the study of variation among such mechanisms requires close analysis of the implementation details of each. Not only is this useful in the pursuit of practical implementations of such tools, but the greater the understanding, the more rationally future design improvements may be made.

References

[1] Jeffrey Pennington, Richard Socher, and Christopher D Manning. GloVe: Global vectors for word representation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 2014.

[2] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 2013.

[3] Ilya Sutskever, Oriol Vinyals, Quoc V. Le. Sequence to Sequence Learning with Neural Networks. *arXiv preprint arXiv:1409.3215,* 2014.

[4] Alex Graves. Sequence Transduction with Recurrent Neural Networks. *arXiv preprint arXiv:1211.3711,* 2014.

[5] Shuohang Wang, Jing Jiang. Machine Comprehension Using Match-LSTM and Answer Pointer. *arXiv preprint arXiv:1608.07905,* 2016.

[6] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473,* 2014.

[7] Rajpurkar P, Zhang J, Lopyrev K, et al. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250,* 2016.

[8] Minh-Thang Luong, Hieu Pham, Christopher D. Manning. Effective Approaches to Attention-based Neural Machine Translation. *Proceedings of the Conference on Empirical Methods in Natural Language Processing,* 2015.

[9] Seo, Minjoon, et al. Bidirectional Attention Flow for Machine Comprehension. *arXiv preprint* arXiv:1611.01603, 2016.