Question Answering with Hybrid Attention Network

Yicheng Li
Department of Electrical Engineering,
Stanford University
ycli@stanford.edu

Xiuye Gu
Department of Computer Science,
Stanford University
xiuyegu@stanford.edu

Abstract

Question Answering (QA) is a Natural Language Processing task that requires the machine to capture interactions between a given passage and a question. We propose a novel hybrid attention network that effectively combines several different attention mechanisms. The output of gated additive attention is piped through a self-attention Layer, and then concatenated with the output of bidirectional attention to form the hybrid attention output, which is processed by a pointer-net layer. Our single model achieves 72.1% F1 and 61.9% EM on the test dataset. The ensemble of seven models achieves 74.9% F1 and 65.3% EM.

1 Introduction

Question Answering (QA) is an important task in Natural Language Processing that has wide applications. In Question Answering, the machine is given a passage (context) and a question, and is asked to predict the answer based on information in the passage. Question Answering is challenging because the model needs to be able to capture complex interactions between the question and the passage and perform reasoning. A major dataset for Question Answering is the Stanford Question Answering Dataset (SQuAD) [1], which contains more than 100000 question-answer pairs. In SQuAD, the answers are guaranteed to appear in the passage, and the model needs to predict the start and end positions of the answer in the passage.

In this work, we propose a hybrid attention neural network, which is inspired by existing high performance models such as r-net [3] and BIDAF [4]. Section 2 gives an overview of those existing models. Section 3 provides a detailed description of our proposed model. Section 4 discusses the performance of our model, factors that affect the performance, and alternative architectures that we implemented.

2 Related Work

Two high-performing models on the SQuAD task are r-net [3] and BIDAF [4]. R-net uses two layers of attention: the first layer is gated additive attention, which produces question-aware passage representations, and the second layer is self-attention, where the passage representation is managed to match itself. A pointer-net layer [5] is also used in r-net to provide better answer span selection. In BIDAF, both context-to-question attention and question-to-context attention are generated to form a bidirectional attention output.

3 Model Architecture

The architecture of the model is shown in Figure 1. The following subsections describe different components of the model.

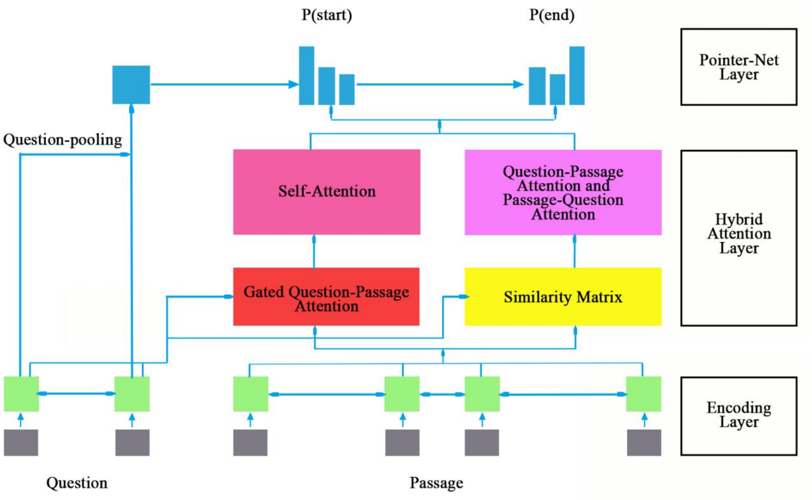

Figure 1: Model architecture overview.

3.1 Embedding and Encoding Layer

The input questions and passages are represented originally as sparse vectors. We convert these tokens to dense representations using pre-trained GloVe vectors of 100 dimensions [2]. Our model then uses a bidirectional GRU layer as the encoder for both the question and the passage. Passage and question word vectors go through the encoding layer and become passage encodings and question encodings. The encodings are used by the hybrid attention layer, which we describe next.

3.2 Hybrid Attention Layer

The hybrid attention layer is the core of our model. It is a mixture of two different attention mechanisms those used in r-net [3] and BIDAF [4].

3.2.1 Gated Question-Passage Attention and Self-Attention

In Figure 1, the left two parts of the hybrid attention layer are the two-layer attention mechanism presented in r-net. Here, the passage and question encodings first go through a gated additive attention module using a GRU:

$$v_t^P = GRU(v_{t-1}^P, gate([u_t^P, c_t]))$$

In the above equation, v_t^P is the hidden state and also output of the GRU. The input at each timestep to the GRU is the gated concatenation of u_t^P, the current passage encoding vector, and the attention-pooling vector c_t, which is computed as:

$$s_j^t = V^T tanh(W_u^Q u_j^Q + W_u^P u_t^P + W_v^P v_{t-1}^P)$$

$$c_t = \sum_{i=1}^{m} softmax(s_i^t) u_i^Q$$

where u^Q and u^P are question and passage encodings, respectively. V, W_u^Q, W_u^P, W_v^P are model parameters to be learned. V is a vector whose dimension is chosen to be one tenth of the encoding size of u^Q and u^P. Increasing the dimension of V will require more memory in computation. At each timestep t, the model takes a token from the passage, and look through the entire question to come up with the attention-pooling vector c_t. The model then concatenates c_t with u_t^P (as introduced in match-LSTM [5]) and applies a gate:

$$g_t = sigmoid(W_g[u_t^P, c_t])$$

$$gate([u_t^P, c_t]) = g_t \circ [u_t^P, c_t]$$

where $\circ$ represents elementwise multiplication. The gated vector is processed by the GRU to output the gated question-passage attention v_t, which is then fed to a self-attention module [3].

$$s_j^t = V^T tanh(W_v^1 v_j^P + W_v^2 v_t^P)$$

$$c_t = \sum_{i=1}^{n} softmax(s_i^t) v_i^P$$

$$g_t = sigmoid(W_g[v_t^P, c_t])$$

$$gate([v_t^P, c_t]) = g_t \circ [v_t^P, c_t]$$

$$h_t^P = BiGRU(h_{t-1}^P, gate([v_t^P, c_t]))$$

In the above equations, V, W_g are the same weights as before, and W_v^1, W_v^2 are new model parameters to be learned. To implement this module, a custom GRU cell is defined, which remembers the entire question encoding matrix and performs additive attention using the input passage encodings and the previous hidden states.

3.2.2 Bidirectional Attention

The right two parts of the hybrid attention layer in Figure 1 represent the bidirectional attention module [4]. In this module, we first compute a similarity score for each passage position i and question position j:

$$S_{ij} = (u_i^P)^T u_j^Q$$

Then we compute the passage-to-question attention a and question-to-passage attention c:

$$\alpha^i = softmax(S_{i,:}) \qquad a_i = \sum_j \alpha_j^i u_j^Q$$

$$m_i = max_j S_{ij} \qquad \beta = softmax(m)$$

$$c = \sum_i \beta_i u_i^P$$

The output vector for each passage position i is a concatenation.

$$b_i = [u_i^P, a_i, u_i^P \circ a_i, u_i^P \circ c]$$

The output of the two attention mechanisms are concatenated to form the output of the entire hybrid attention layer.

$$h_i^{attn} = [h_i^P, b_i]$$

3.3 Pointer-net Layer

This concatenated output of the hybrid attention layer is fed to a pointer-net layer [3], which outputs the probability distribution for the start and end positions. In the pointer-net layer, a GRU is used so that the end position distribution is conditioned on the start position distribution.

$$s_i^1 = V^T tanh(W_h^P h_i^{attn} + W_h^a h_0^a) \qquad a_i^1 = softmax(s_i^1)$$

a_i^1 is the probability distribution of the start position. We then use the GRU to generate the end position distribution based on the start position distribution. Because we only need the start and end positions, we use the boundary model in [5], and apply the GRU for one step:

$$c_1 = \sum_{i=1}^{n} a_i^1 h_i^{attn}$$

$$h_1^a = GRU(h_0^a, c_1)$$

$$s_i^2 = V^T tanh(W_h^P h_i^{attn} + W_h^a h_1^a)$$

$$a_i^2 = softmax(s_i^2)$$

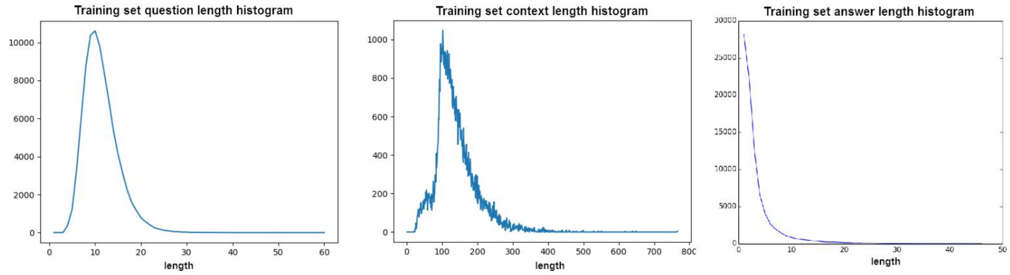

Figure 2: Histograms for question, context, answer lengths in training set.

a_i^2 is the probability distribution of the end position.

The initial state h_0^a of this GRU is the question-pooling vector, which is a dense representation of the entire question, as described in r-net [3].

$$s_i^{pooling} = V^T tanh(W_u^Q u_i^Q + W_V^Q V_r^Q)$$

$$a_i^{pooling} = softmax(s_i^{pooling}) \qquad h_0^a = \sum_i a_i^{pooling} u_i^Q$$

In implementation, V^T is the same as in the gated question-passage attention and self-attention models described above. W_u^Q, W_V^Q, V_r^Q are new parameters to be learned in this layer. After obtaining the start and end distributions a^1, a^2, we predict start and end positions to be the positions with largest probability by taking the argmax of the distributions.

4 Results and Discussion

4.1 Implementation Details

Our model is trained and evaluated on the SQuAD dataset. We generated histograms of question, passage and answer lengths in the training set, as shown in Figure 2. It can be seen that the most questions are no longer than 25 words, most passages are no longer than 400 words, and most answers are no longer than 20 words. Because the model processes data in batches, it needs a certain maximum length for questions and contexts, which it pads the input to reach if the input is shorter than that. Allowing an unnecessarily long length would be a waste of computational resources. So, in training, we allowed a maximum question length of 30 words, and a maximum context length of 450 words. We used Adam optimizer with a learning rate decaying from 0.001 to 0.00001, a dropout rate of 0.15, and pretrained GloVe vectors of 100 dimensions. We clip the gradient if the norm reaches 5.

4.2 Results and Effect of Components

We use F1 scores and Exact Match (EM) to evaluate performance of the model. The F1 score is the harmonic average of Precision and Recall, and measures the amount of overlap between the ground truth answer and the predicted answer. The EM score measures whether the true answer and the predicted answer are exactly the same. The overall F1 and EM scores are the average over all questions. The performance of our model and comparison with BIDAF and r-net are summarized in Table 1. The dev scores are lower than the test scores because they are measured by comparing the predicted answer with one ground truth answer only, while in the actual test there are more than one correct answers provided. Although our model uses mechanisms introduced in BIDAF and r-net, the performance is not as good as those two models, which could be explained by the following facts:

1. Our model did not use character level embeddings. 2. The gated question-passage attention module in our model uses one-directional GRU, where r-net uses bidirectional GRU. 3. We only used one layer of bidirectional GRU for encoding passage and question, while r-net uses three layers [3]. 4. In the bidirectional attention module, we used multiplication to compute the similarity matrix S,

Table 1: Performance comparison with other methods

Model	Dev F1 (%)	Dev EM (%)	Test F1 (%)	Test EM (%)
Our model (single)	65.9	51.0	72.1	61.9
Our model (ensemble)	68.2	61.0	74.9	65.3
BIDAF			77.3	68.0
r-net			84.2	76.5

Table 2: Dev F1 and EM improvement for each component added

Change in model	Dev F1 (%)	Dev EM (%)
Baseline (one basic attention layer)	40	29
Add self-attention	54	38
Add pointer-net layer	57	41
Change basic attention to gated additive attention	65	50
Add bidirectional attention to form hybrid attention	66	51

while BIDAF uses a learned weight vector to compute the similarity between question and passage [4]. 5. We did not use a modeling layer as in BIDAF.

We also provide a breakdown of important components in our model and their effects in improving model performance in Table 2. Starting from the baseline (simple one-directional attention model), we found that the biggest improvements came from adding self-attention layer (dev F1 40% to 54%) and changing the baseline attention layer to gated additive attention (dev F1 57% to 65%). Adding the pointer-net layer and constructing a hybrid attention layer were also helpful. We also tried predicting start and end positions by maximizing the product of start position probability and end position probability subject to the constraint that the end position is within 0 to 15 words after the start position, but did not get further improvements, possibly because the pointer-net layer already took this relationship into account.

4.3 Attention Analysis

In this subsection, we provide visualizations for one question. The passage is: ”in the centre of basel, the first major city in the course of the stream, is located the ”rhine knee”; this is a major bend, where the overall direction of the rhine changes from west to north. here the high rhine ends. legally, the central bridge is the boundary between high and upper rhine. the river now flows north as upper rhine through the upper rhine plain, which is about 300 km long and up to 40 km wide. the most important tributaries in this area are the ill below of strasbourg, the neckar in mannheim and the main across from mainz. in mainz, the rhine leaves the upper rhine valley and flows through the mainz basin.”. The question is: ”How long is the upper rhine plain?”. The answer is ”300 km long” and the model answered this question correctly. Figure 3(a) shows the gated additive attention intensity for each pair of question token and passage token. The question has 8 tokens (shown as rows), and the passage has 133 tokens (shown as columns). It can be seen that although the highest intensity comes from ”upper rhine plain”'s attention to ”upper rhine” and ”high rhine”, the attention from ”long” to ”300 km long” is also very strong, which enables the model to find the correct answer. Interestingly, the attention from ”long” to ”about” and ”40 km wide” are also strong, which means the model has considered answering ”about 300 km long” (a decent answer but not the best) or ”40 km wide” (a wrong answer that is very similar to the true answer in format), but successfully avoided them.

Figure 3(b) shows the bidirectional attention similarity matrix S_{ij}. It can be seen that this second attention mechanism we added to the side of our model captures similar information as gated additive attention, but focuses more on the useful information (the signal-to-noise ratio of this one is higher than Figure 3(a)). For example, now ”long” no longer attends to ”40 km wide” intensively. This might be a reason why adding bidirectional attention mechanism helps improve the model performance.

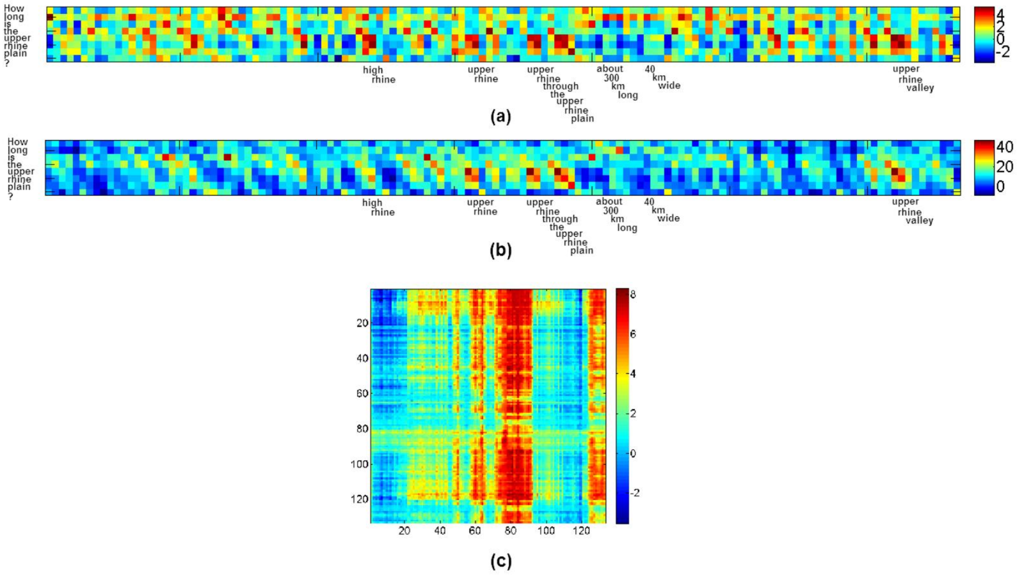

Figure 3: Attention visualization for question ”How long is the upper rhine plain ?”. Zoom in to see details. (a) gated additive attention. (b) bidirectional attention similarity matrix. (c) self-attention.

Figure 3(c) shows the self-attention intensity $s_j^t = V^T tanh(W_v^1 v_j^P + W_v^2 v_t^P)$. This is where the question-aware passage representation output by gated additive attention attends to itself. It can be seen that here the passage representation focuses on the 75th-90th words in itself, which are ”upper rhine plain, which is about 300 km long and up to 40 km wide”. This shows that after reading and remembering the question, the model looks at the context again and knows to focus on this part, which is indeed where the answer comes from. Thus the attention mechanisms are quite reasonable, interpretable and effective.

4.4 Error Analysis

In this subsection, we categorize some of the most typical mistakes our model makes.

1. The predicted answer says the same thing as the true answer but with slightly longer or shorter representation. Example:
QUESTION: what type of materials inside the cabin were removed to help prevent more fire hazards in the future?
TRUE ANSWER: flammable cabin and space suit materials
PREDICTED ANSWER: flammable cabin and space suit
This type of mistakes is not serious. The model's answer is actually good.

2. The predicted answer contains the true answer but also contains some irrelevant words that a human answerer would not add. Example:
QUESTION: when forces are acting on an extended body, what do you need to account for motion effects?
TRUE ANSWER: respective lines of application
PREDICTED ANSWER: their respective lines of application must also be specified in order
In this example, the model should not say ”must also be specified in order”. Although the information it gives is correct, the output is not natural. Humans are unlikely to answer this question in this way. Since we want the model to behave like humans, this type of mistake is unacceptable.

3. In addition to the true answer, the predicted answer also contains something that should clearly be excluded. Example:
QUESTION: in which continent besides asia were major gains made by the british empire in the late 19th century ?

TRUE ANSWER: middle east
PREDICTED ANSWER: asia and the middle east
Here the model says ”asia”, which is clearly wrong because the question says ”besides asia”. This type of mistake is unacceptable.

4. The model fails to realize that two expressions are equivalent. Example:
CONTEXT: during the period in which the negotiations were being conducted, tesla said that efforts had been made to steal the invention. his room had been entered and his papers had been scrutinized , but the thieves, or spies, left empty-handed.
QUESTION: according to tesla what had been gone over by the thieves, or spies who entered his room?
TRUE ANSWER: his papers
PREDICTED ANSWER: empty-handed
Here the model fails to realize that ”gone over” is equivalent to ”scrutinized”. The model may think that ”gone over” means ”left”. This type of mistake can possibly be reduced by training the word vectors or using higher-dimension word vectors.

5. The model fails because it does not use grammar information. Example:
for a long time , number theory in general, and the study of prime numbers in particular, was seen as the canonical example of pure mathematics.
QUESTION: besides the study of prime numbers, what general theory was considered the official example of pure mathematics?
TRUE ANSWER: number theory
PREDICTED ANSWER: canonical
Here, if the model knows that the answer has to be a noun, and should be the subject of ”was seen”, it would not have made this mistake. Adding part-of-speech features and subject-verb-object relationships may help solve such problems.

4.5 Experiments on Other Possible Architectures

In addition to the final model architecture presented above, we also implemented and tested a number of alternative architectures. We now give an overview of these architectures and their performance.

4.5.1 Hybrid Attention Model with Coattention

A variation of the model described in Section 3 is to replace the bidirectional attention module with coattention, as shown in Figure 4.

The coattention layer, as introduced in [6], first applies a non-linear projection to the question encodings u^Q:

$$u^{Q'} = tanh(Wu^Q + b)$$

The coattention layer then adds sentinel vectors u_0^P, u_0^Q to the passage and question encodings to enable the model to attend to none of the token encodings, and then computes the affinity matrix L from the encodings u^P, u^Q.

$$L = [u^P, u_0^P]^T [u^{Q'}, u_0^Q]$$

where u^P, u^Q are passage and question encodings, respectively. Then, Context-to-Question Attention and Question-to-Context Attention are computed from matrix L. The Second-Layer Attention is computed from the Question-to-Context Attention, and the output is generated by a bidirectional LSTM whose input is the concatenation of Second-Layer Attention and Context-to-Question Attention.

This alternative model then concatenates the coattention output with self-attention output to form the output of the hybrid attention layer. This model achieves a dev F1 score of 0.647 and a dev EM score of 0.496. This shows that in the hybrid attention architecture, coattention is less helpful than bidirectional attention, possibly because coattention is very different from the r-net attention mechanism, and the model has difficulty learning to combine these two mechanisms.

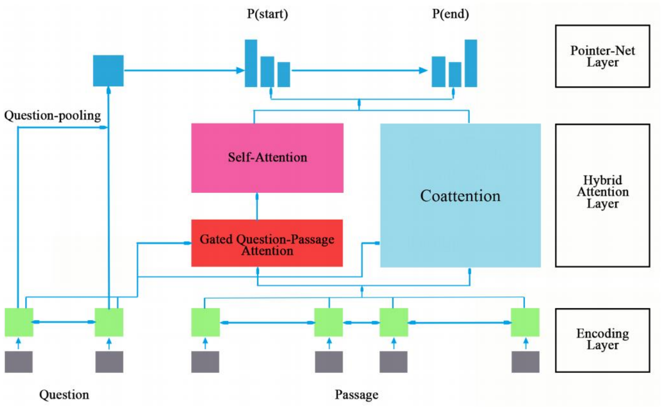

Figure 4: Hybrid attention model with coattention.

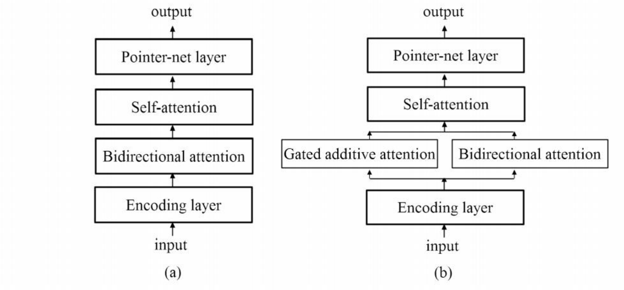

Figure 5: Two alternative architectures.

4.5.2 Replacing Gated Additive Attention with Bidirectional Attention

Another variation is to directly replace the gated additive attention layer with bidirectional attention, as shown in Figure 5(a). This model achieves a dev F1 score of 0.651 and a dev EM score of 0.494, which is not as good as the hybrid model proposed in Section 3, possibly because it only uses one attention mechanism and is not as expressive as the hybrid model.

4.5.3 Alternative Ways of Combining Bidirectional Attention and R-net

In addition to concatenating bidirectional attention output with self-attention output, as described in Section 3, we also tried concatenating it with gated additive attention, as shown in Figure 5(b). This model achieves a dev F1 score of 0.570 and a dev EM score of 0.421, which is far worse than the previous hybrid model, possibly because the self-attention layer would have difficulty learning to utilize these two first-layer attention modules because they are very different - one is controlled by a gate and the other is itself a concatenation of attention in two directions.

5 Conclusion and Future Work

In conclusion, we proposed a novel hybrid attention architecture that combines attention gated additive attention, bidirectional attention and self-attention. Our single model achieves 72.1% F1 and 61.9% EM on the test set. Attention analysis shows that our model is effective in understanding the interactions between question and passage. Future work includes better interpretation of the function of key modules in the architecture. It might also be helpful to have the model identify the subject, verb and object of a sentence, as suggested by the error analysis section.

Acknowledgments

We thank the instructors and staff of CS224n for providing a great course experience.

References

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, ”Squad: 100,000+ questions for machine comprehension of text,” *arXiv preprint* arXiv:1606.05250, 2016.

[2] J. Pennington, R. Socher, and C. D. Manning, Glove: Global vectors for word representation, in *EMNLP*, vol. 14, pp. 15321543, 2014.

[3] W. Wang, N. Yang, F. Wei, B. Chang and M. Zhou, ”Gated Self-Matching Networks for Reading Comprehension and Question Answering,” in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 189-198.

[4] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, Bidirectional attention flow for machine comprehension, *arXiv preprint* arXiv:1611.01603, 2016.

[5] S. Wang and J. Jiang. ”Machine comprehension using match-lstm and answer pointer,” *arXiv preprint* arXiv:1608.07905, 2016b.

[6] C. Xiong, V. Zhong, and R. Socher, Dynamic coattention networks for question answering, *arXiv preprint* arXiv:1611.01604, 2016.