Question Answering

Bowen Deng
Department of Statistics
Stanford University
Codalab username: baolidakai
bdeng2@stanford.edu

Abstract

Question answering has been recently a hot topic, especially with application on virtual assistant. The performance of a model on question answering task is also an important benchmark for evaluation and understanding the model better. The maturity of attention model is very important in achieving good performance for a question answering system. This project re-implements some of the ideas in published paper, and achieved an F1 of 67% and EM of 55%, on the test dataset.

1 Introduction

Stanford Question Answering Dataset (SQuAD) is a dataset of reading comprehension. The data for the model is a tuple of a paragraph, a question, a start position, and end position, meaning the question is answered by the paragraph between the start position and end position. Solving this task well is a strong indicator of the performance of a new algorithm or architecture, for academia; and also has practical implications, such as building a virtual assistant such as Siri, Google Home, and Alexa, or useful in web answers feature in Google search.

The task of an algorithm is to predict the start position and end position for a paragraph (context), question pair. And the algorithm is evaluated with two metrics: EM, the percentage of exact match; and F1, the harmonic mean of precision and recall.

In this project, several attention models are implemented on top of the baseline model with basic attention: bidirectional attention, coattention, and self-attention. The best performance is obtained with extension with coattention layer + self-attention layer. Example output of the algorithm is shown and analyzed.

2 Approach

2.1 Baseline

The paragraph (context) is represented as a sequence of word embeddings $x_1, \dots, x_N \in \mathbb{R}^d$, and similarly the question is represented as $y_1, \dots, y_M \in \mathbb{R}^d$. The word embedding used in this project is pretrained GloVe embeddings. Both inputs are feed through a 1-layer bi-GRU with shared weights.

The next layer is a basic attention layer, which gives the attention output $\mathbf{a} = \alpha_i^T q$, where

$$\alpha_i = softmax(c_i^T q)$$

The output layer is a fully connected layer with input c, a,

$$b' = ReLu(W_1 c_i + W_2 a_i + b_1)$$

To compute the probability distribution of start position and end position, we use the softmax of a downprojection layer:

$$p^{start} = softmax(W_3^T b' + b_2), p^{end} = softmax(W_4^T b' + b_3)$$

2.2 Bidirectional Attention

Given the context $c_1, \ldots, c_N$, and question $q_1, \ldots, q_M$, we first compute the similarity matrix

$$S_{ij} = w_1^T c_i + w_2^T q_j + w_3^T c_i \circ q_j$$

The next step is a Context-to-Question attention, which is same as the attention layer in baseline, except we are using S as the input.
We also apply a Question-to-Context attention, by first getting the row max

$$m_i = \max_j S_{ij}$$

then compute the weighted sum of context:

$$c' = softmax(m)^T c$$

Finally, we blend context c, Context-to-Question attention a, $c \circ a$, and $c \circ c'$, as the attention output. See the figure below for a visualization [3].

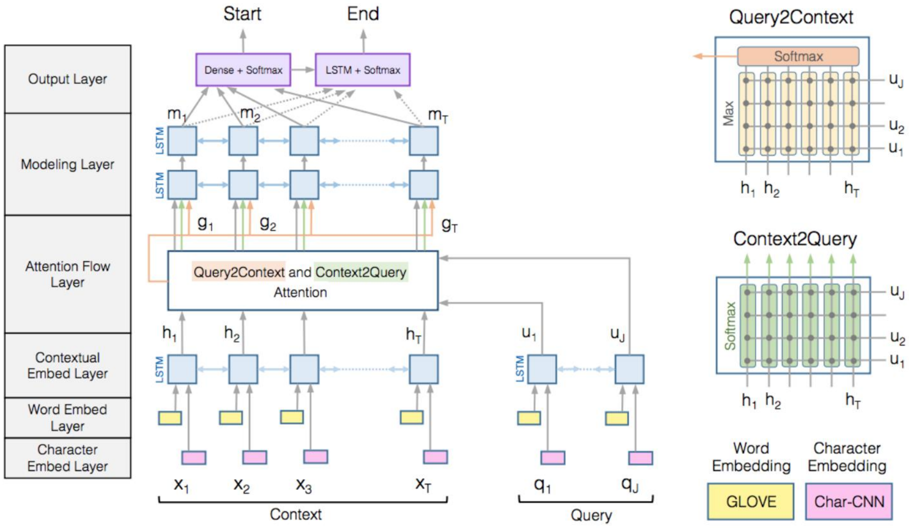


Figure 1: Architecture of Bidaf

2.3 Coattention

The major difference of coattention model, compared with bidirectional attention, is the use of a second-level attention layer, which attends over representations that are attention. After computing α, a, and b, it introduces a new layer:

$$s = \alpha^T b$$

And finally concatenate s and a, as the input to a bidirection LSTM layer, whose output is the attention output.
See the figure below for a visualization [7].

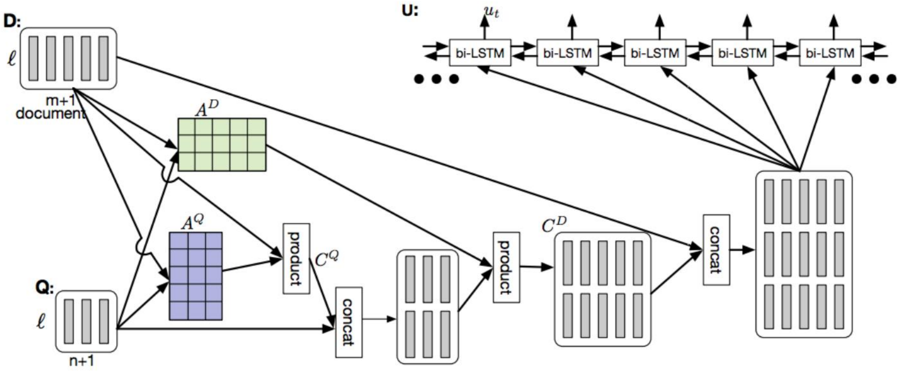


Figure 2: Architecture of Coattention Layer

2.4 Self Attention

The idea of self attention is to insert an additional attention layer, which takes a sequence $x_1, \ldots, x_N$, as input. And compute

$$e_{i,j} = v^T tanh(W_1 v_j + W_2 v_i)$$

then compute the weighted sum of softmax of $e_{i,j}$.
We insert this self attention layer to baseline, bidirection attention, and coattention layer.
See the figure below for a visualization [6].

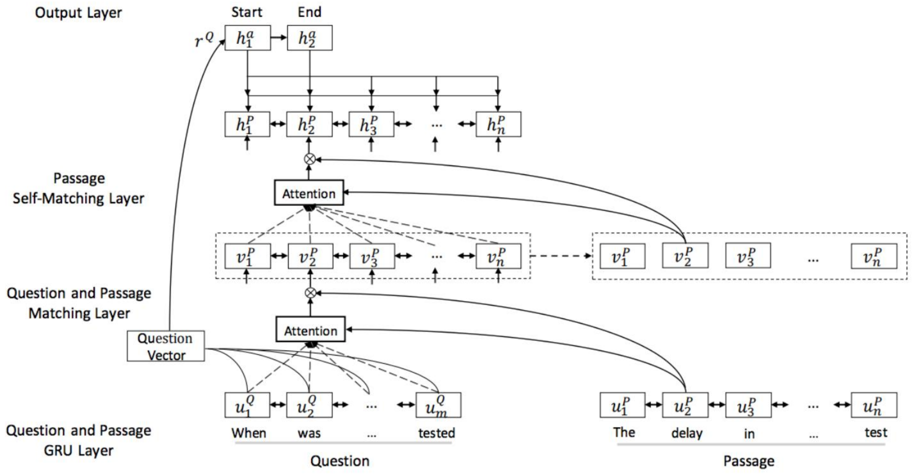


Figure 3: Architecture of Self-attention Layer

2.5 Answer Pointer Layer

Motivated by the Pointer Net introduced by [4], answer pointer layer with a sequence model and boundary model is implemented, which did not gain too much improvement in this project's implementation.
See the figure below for a visualization [5].

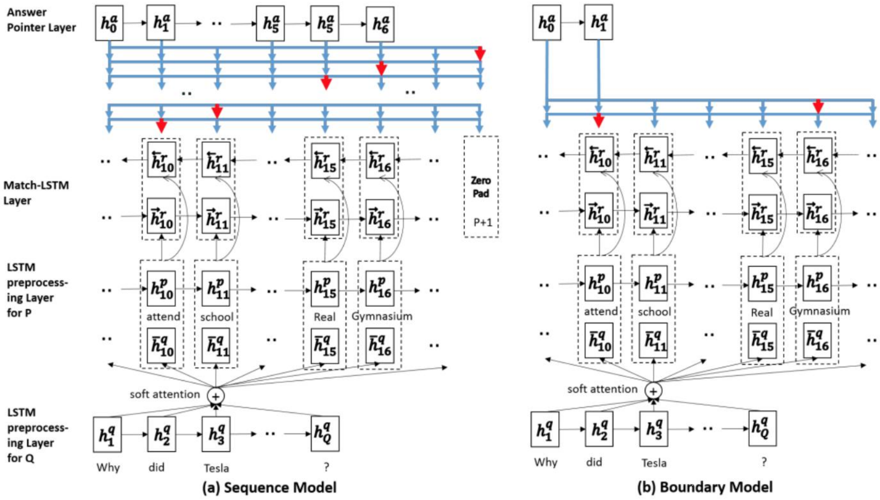


Figure 4: Architecture of Sequence model and Boundary model

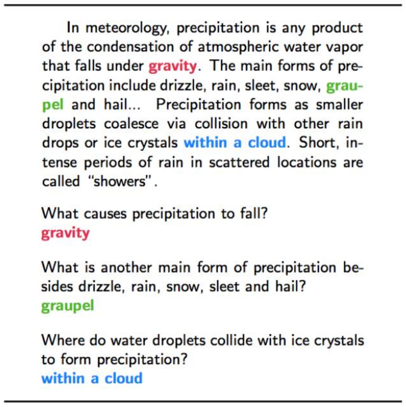
2.6 Prediction

The baseline approach to predict the start position, and end position, given the distribution of them, is to pick the start position with maximum probability, and end position with the maximum probability. However, it is possible that start position is on the right of the end position, thus, we pick two positions such that start position is at the left of right position, whose product of probability is maximized.

3 Experiment

3.1 Data

The SQuAD dataset [2] is used for model development, training, and evaluation. The passages are more than 20k single paragraph from Wikipedia articles, associated with around 5 questions. For evaluation, the data has been split into 80% training set, 10% dev set, and 10% hidden test set.
See an example of a passage with associated questions below:



In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

Figure 5: Example passage and associated questions

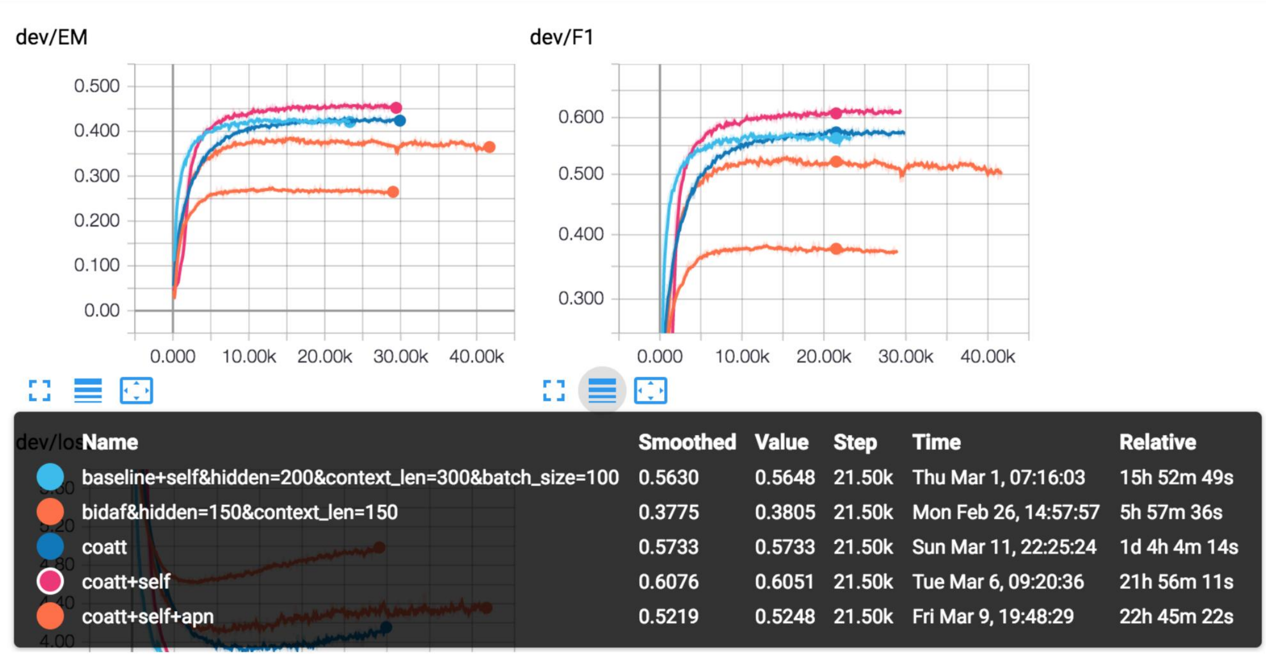
3.2 Experiment Settings

We use pretrained GloVe word embeddings [1], and impute missing words with 0.
Various size of hyperparameters: hidden layer sizes, learning rate, batch size, dropout rate are used, for each different combinations of methods.
The methods tried are no attention, baseline attention model, bidirectional attention, coattention. In addition, each of the models are combined with/without self attention, and with/without answer pointer network. Each of the models are tuned with different hyperparameters.

3.3 Evaluation

The two evaluation metrics used to compare across different models is the exact match score, and F1 score with token-level. The exact match score is defined as the percentage of exact match for start and end position. The F1 score for word-level, is a less strict score, that compute the harmonic mean of precision and recall.
We use evaluation dataset for hyperparameter tuning and model selection, and submit the best model to the hidden test dataset.

3.4 Results



Name	Smoothed	Value	Step	Time	Relative
● baseline+self&hidden=200&context_len=300&batch_size=100	0.5630	0.5648	21.50k	Thu Mar 1, 07:16:03	15h 52m 49s
● bidaf&hidden=150&context_len=150	0.3775	0.3805	21.50k	Mon Feb 26, 14:57:57	5h 57m 36s
● coatt	0.5733	0.5733	21.50k	Sun Mar 11, 22:25:24	1d 4h 4m 14s
● coatt+self	0.6076	0.6051	21.50k	Tue Mar 6, 09:20:36	21h 56m 11s
● coatt+self+apn	0.5219	0.5248	21.50k	Fri Mar 9, 19:48:29	22h 45m 22s

Figure 6: EM and F1 score on development dataset across different settings

Part of the settings with their development metrics are shown above. The best model for both EM score and F1 score is coattention, combined with self attention layer, without using answer pointer layer. The corresponding F1 score on test dataset is 67%, and EM score 55%. The best hyperparameters picked are:

Hyperparameter	Value
Hidden layer size	200
Context length	300
Mini-batch size	100
Learning rate	0.001
Dropout rate	0.15

Table 1: Hyperparamters for coattention + self-attention model

4 Conclusion

In this project, we implemented different attention models and answer pointer layer, to solve the question answering problem. The models are trained and evaluated on SQuAD dataset. The model with best test F1 score is 67%, with exact match score 55%.

One possible future direction is to re-train the word embedding, which is fixed in this project. Another possible extension is to model the conditional end position, given the start position, hence, improve the robustness of the model.

References

[1] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. 2014.

[2] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. *CoRR*, abs/1606.05250, 2016.

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

[4] O. Vinyals, M. Fortunato, and N. Jaitly. Pointer Networks. *ArXiv e-prints*, June 2015.

[5] Shuohang Wang and Jing Jiang. Machine comprehension using match-lstm and answer pointer. *CoRR*, abs/1608.07905, 2016.

[6] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. pages 189–198, 01 2017.

[7] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. *CoRR*, abs/1611.01604, 2016.