Question Answering System with the Dynamic Coattention Network

Yi Sun*
Department of Statistics
ysun4@stanford.edu

Abstract

For the default project of CS 224n course, we basically did a reimplementation of the Dynamic Coattention Network. We built the architecture from the baseline model provided, tuned the network and tried different regularization strategies. Eventually, we are able to get F1 score 51.705 and EM 43.457 on test set.

1 Introduction

Despite recent popularity and success of Natural Language Processing(NLP), the task of understanding the meaning of a text and answering questions remains to be a challenge for NLP researchers. One benchmark dataset for such reading comprehension problem is Stanford Question Answering Dataset(SQuAD)[1]. Though SQuAD established in less than two years, many significant break through has been made to provide amazing results in this field.

In this paper, we outline our methods in developing the neural model and experiment with different different tunnning parameters.

2 Background/ Problem Setting

The SQuAD database is consists of paragraphs from Wikipedia and a question and its answer about the paragraph from Amazon Mechanical Turk. There are 100K questions in total. Our goal is to answer the question correctly given the paragraph and the question.

We evaluate the results of our predictions using F1 and Exact Match(EM) score. EM score is a strict binary measure that will return 1 if the predicted answer is the same as the true answer(complete match) and 0 otherwise. F1 score is the harmonic mean of precision and recall, where roughly, precision is how much the predicted answer falls into the true answer and recall the the proportion of true answer being included in the predicted answer.

The final score of our model is the mean of F1 and EM score of each question.

3 Approach Model

Our model consists of three main parts: a RNN encoder layer, a co-attention layer and an output layer. We use following notation setting in the section.

For every SQuAD example (consists of question, context, answer), contexts and questions will be represented by a sequence of word embeddings of dimension d, context is represented by $x_1, ..., x_N \in \mathbb{R}^d$ and question represented by $y_1, ..., y_M \in \mathbb{R}^d$.

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

3.1 RNN encoder layer

The RNN encoder layer encodes both sequences of word embedding of contexts and questions to hidden states.

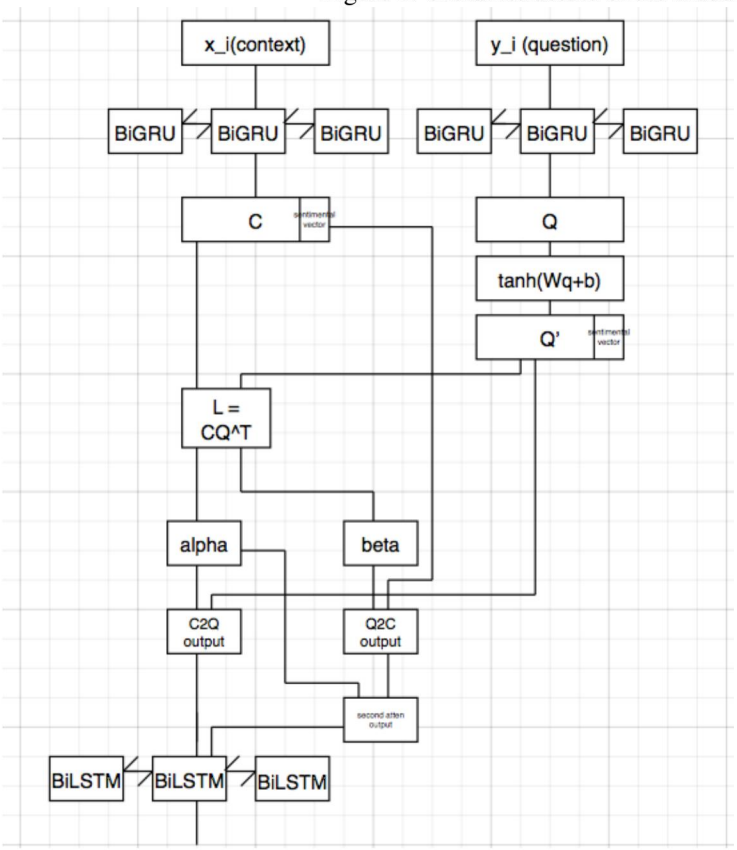
We used the code in the baseline which applies a bidirectional GRU on the word embeddings of context and question.

Let h be the resulting hidden state from GRU. Then we concatenate the forward hidden state and the backward hidden state to generate the hidden state for context and question. And obtain $c_1, ..., c_N \in \mathbb{R}^{2h}$ for context hidden states and $q_1, ..., q_M \in \mathbb{R}^{2h}$ for question hidden states.

3.2 Coattention

The idea of this layer is from the Dynamic Coattention Network[2], which involve a second-level attention layer, which is attending over the outputs of first step attention. This layer can be decomposed into four steps. The overall graph is shown in Figure1. For $c_1, ..., c_N \in \mathbb{R}^{2h}$ and $q_1, ..., q_M \in \mathbb{R}^{2h}$

Figure 1: overall structure of the model



as inout from last layer.

The first is to introduce a linear layer with tanh() as activation function to obtain projected question hidden states. We obtain $q'_1, ..., q'_M \in \mathbb{R}^{2h}$ from

$$q'_j = tanh(Wq_j + b)$$

where W as weight matrix and b as bias.

Then we introduce a sentimental vector $c_\emptyset$ and $q'_\emptyset$ as trainable variable to context and question hidden state which allow us avoid error in attaining to none of the provided state.

We got C = $\{c_1, ..., c_N, c_\emptyset\}$ and Q= $\{q'_1, ..., q'_M, q'_\emptyset\}$, where $C \in \mathbb{R}^{(N+1)\times 2h}$, $Q \in \mathbb{R}^{(M+1)\times 2h}$

Then our first-level attention consists of two attentions, ConText-To-Question(C2Q) and Question-to-Context(Q2C) Attention. We obtain affinity matrix $L \in \mathbb{R}^{(N+1)\times(M+1)}$ by

$$L = CQ^T$$

In C2Q attention, we get C2Q attention output a by:

$$\alpha = softmax(L)$$

along the first dimension of L, thus $\alpha \in \mathbb{R}^{(N+1)\times(M+1)}$

Then

$$a = \alpha Q$$

where $a \in \mathbb{R}^{(N+1)\times 2h}$

In Q2C layer, similarly

$$\beta = softmax(L)$$

softmax along second dimension, thus $\beta \in \mathbb{R}^{(M+1)\times(N+1)}$ Then

$$b = \beta C$$

where $b \in \mathbb{R}^{(M+1)\times 2h}$

Then in the second level attention, we get second-level attention output S as:

$$S = \alpha b$$

Thus $S \in \mathbb{R}^{(N+1)\times 2h}$

We feed a bidirectional LSTM with the concatenation of $[S; a]$ and obtain resulting $\{u_1, ..., u_N\}$

3.3 output layer

We obtain U $\in \mathbb{R}^{(N+1)\times 4h}$ from last Dynamic Coattention layer.

Find U' by a fully connected layer with ReLU,

$$U' = ReLU(UW_{FC} + b_{FC})$$

Then we compute the result probability p_{start} and p_{end} aw follows, let $k \in start, end$

$$p^k = softmax(W_k U' + b_k)$$

Where W_k and b_k are weight matrix and bias vector.

3.4 Loss and Prediction

The loss is the sum of cross entropy loss for the start and end locations. For a single outcome:

$$loss = -logp^{start}(i_{start}) - -logp^{end}(i_{end})$$

where i_{start} and i_{end} are the start and end location. The overall loss is the mean of single loss.

We predict the answer interval (l^{start}, l^{end}) by simply take the argmax of p^{start} and p^{end}:

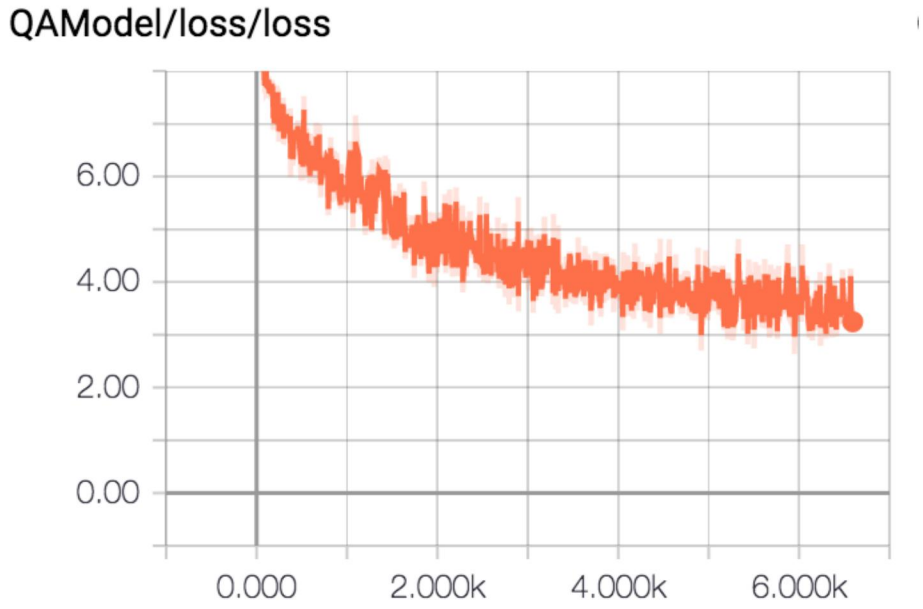
$$l^{start} = argmax(p^{start})$$

$$l^{end} = argmax(p^{end})$$

4 Experiments

We try to improve model performance by adjusting following parameter tuning and function used. The resulting loss function is in Figure 2.

Figure 2: The loss function in model



4.1 regularization function

In the baseline, we mainly use dropout as way of regularization, but there are also other ways of regularization. We tried L2-regularization and L1-regularization as additional regularizer. However, there is not much improvement. The reason I guess is that the both dropout and L2-regularization both serves as regularization method and the parameter is very well-tuned. As using L-2 regularization will introduction one more tuning parameter, but in the experiment the process of tuning parameter is very time-consuming.

4.2 batch size

The batch size is the number of examples the model look before making one weight update. Thus, if the training signal becomes to high , we should increase the batch size. However, if batch size become too small it incurs large overhead, and will slows down training, batch size too large incurs OOM. We have tried 80, 100, 120, it turns out 120 will lead to OOM, 100 will sometimes leads to OOM, so we use 80 as batch size.

4.3 learning rate

The learning rate decides the performance of convergence, as we use values 0.001, 0.005, 0.0008 and 0.0015. For learning rate too large (e.g. 0.005) the model simply do not converge. If we choose small learning rate, by definition, it would take longer to reach optimum point. Thus, we use the default 0.001 which does not work as perfect as we expect but would not suffer from divergence for being too large.

4.4 context_len

The context_len is the maximum context lengths allow during training. If the input has length greater than this, it will be discarded. This parameter by its definition decrease the number of training example as we decrease it, but it can boost the speed of training process. We use 500 instead of 600 just to get some speed up without loss too much information.

4.5 Dropout rate

The dropout rate controls the degree of regularization and also the speed of training. Since the dropout rate control the degree of regularization. We only increase it if we find there is great discrepancy between the loss function on training set and dev set. We found dropout rate to 0.001 works fine.

5 Discussion

5.1 Error Analysis

Error Analysis Example 1

CONTEXT: southern california is home to many major business districts . central business districts (cbd) include downtown los angeles , downtown san diego , downtown san bernardino , downtown bakersfield , south coast metro and downtown riverside .

QUESTION: what is the only district in the cbd to not have " downtown " in it 's name ?

TRUE ANSWER: south coast metro

PREDICTED ANSWER: central business districts

ANALYSIS: The problem is the model misclassified central business district as a specific business district instead of a general reference to a group. This is caused by the model does not notice the key word "include", it seems the model does not see the information after the word, so the model can be improvement in word embedding.

Error Analysis Example 2

CONTEXT: to remedy the causes of the fire , changes were made in the block ii spacecraft and operational procedures , the most important of which were use of a _nitrogen/oxygen_ mixture instead of pure oxygen before and during launch , and removal of flammable cabin and space suit materials . the block ii design already called for replacement of the block i _plug-type_ hatch cover with a quick-release , outward opening door . nasa discontinued the manned block i program , using the block i spacecraft only for unmanned saturn v flights . crew members would also exclusively wear modified , fire-resistant block ii space suits , and would be designated by the block ii titles , regardless of whether a lm was present on the flight or not.

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: space suit materials

ANALYSIS: The model just answer the question half right. It answer the part "space suit material" right but ignore the "flammable cabin". The model ignore a very key word "and" so it miss the second part of the true answer. Also, I guess the model predict the second half of the true answer instead of the first half is because the topic of context us about space. I guess we can give model some additional information to deal with word like "and", "but", "or".

Error Analysis Example 3

CONTEXT: abc also owns the times square studios at 1500 broadway on land in times square owned by a development fund for the 42nd street project ; opened in 1999 , good morning america and nightline are broadcast from this particular facility . abc news has premises a little further on west 66th street , in a six-story building occupying a 196 feet (60 m) _ 379 feet (116 m) plot at _121-135_ west end avenue. the block of west end avenue housing the abc news building was renamed peter jennings way in 2006 in honor of the recently deceased longtime abc news chief anchor and anchor of world news tonight .

QUESTION: a block of west end avenue that houses an abc news building was renamed for what abc anchor ?

TRUE ANSWER: peter jennings

PREDICTED ANSWER: world news tonight

ANALYSIS: The predicted answer is not people's name but a name of news, which is partially correct in category. I guess it is because the question ask about "abc anchor" and name "world news tonight" is just near the two "anchor" words in the context, so maybe it puts some weight to disrupt the prediction. This means the model does not get the meaning of the sentence correctly. I guess it suggest we need to train more times to let model learn or have some scheme for the model to escape such trap.

5.2 Conclusion

Co-attention works well with our dataset and greatly improve the result comparing to the baseline. The final F1 and EM score we getis 51.705 and 43.457 respectively. Comparing to the result from baseline, where both models are trained with similar amount of samples(7 epochs) and new model with co-attention layer reaches 8 and 9 points increase in dev set in F1 and EM scores respectively.

5.3 Future Improvement

The project reach great result comparing to our baseline but there is still much room for improvement. The character level CNN should be another great boost in performance of this model given its popularity. Also, more parameters is not well tuned because tuning parameter is time-consuming and the change of one parameter will lead to a shift in the optimal value of other parameters.

References

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

[2] Caiming Xiong, Victor Zhong, and Richard Socher. Dynamic coattention networks for question answering. arXiv preprint arXiv:1611.01604, 2016 Systems 7, pp. 609-616. Cambridge, MA: MIT Press.