Bi-Directional Attention & Self Attention for SQUAD

Li Deng
Department of Computer Science
Stanford University
Stanford, 94305
dengl11@stanford.edu

Zhiling Huang
Department of Computer Science
Stanford University
Stanford, 94305
zhiling@stanford.edu

Abstract

In this paper, we used bi-directional recurrent neural network (LSTM, GRU), self attention, bi-directional attention, combined attention, ensemble etc. to solve a reading comprehension problem, SQUAD. We reached F1 score **0.756** and EM **0.649**.

1 The Stanford Question Answering Dataset

The Stanford Question Answering Dataset[1] is a reading comprehension data set. The data includes context paragraphs, questions for each context paragraph and answers to the questions, where answers are substrings of the context paragraphs.

2 Metrics

$$EM = N_{exact\_match}/N_{total\_predictions}$$

$$precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

3 Data Analysis

To better understand the data and the problem, we first analyzed the dev data to see whether there is any obvious pattern that we can take advantage of during training.

We first noticed that most($> 99\%$) of the true answer has a length that is shorter than 15 words. This means when we make predictions we can assign less probability to potential answer candidates that are longer than 15 words, or totally ignore them.

The second observation is that most($> 99\%$) of the true answer appears within 300 words from the start of the context paragraphs. We can safely reduce the size of the model by cutting off the second half of context that is longer than 300 words.

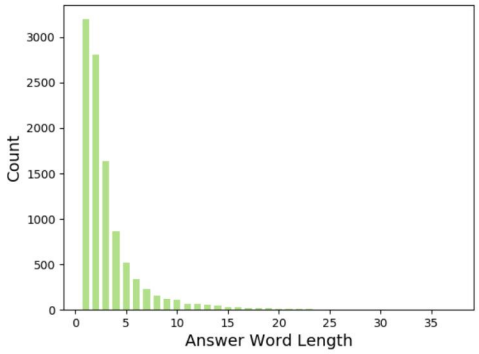


Figure 1: Answer word length

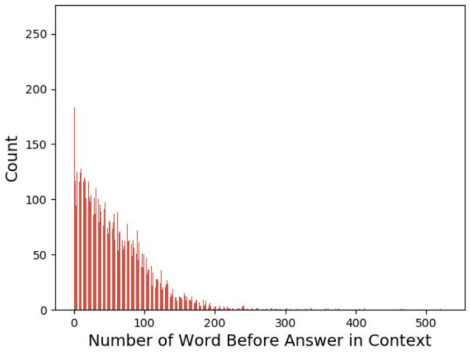


Figure 2: Number of word before answer

4 Data Preprocessing and Extra Features

We represent each word in context paragraphs and questions with GloVe word vectors of length 100.

In order to batch context paragraphs and questions of various lengths, we padded all the context paragraphs to 300 and all the questions to max question length, which is 30. The padded index will be masked and ignored in the model.

We also calculated Inverse Document Frequency for each word in the training data. The idea is that the less common the word is in all context paragraphs, the more important the word is in the specific context paragraph that it appears in. For instance, 'the', 'in' are very common words and they are in almost every context paragraph, but they usually are not strong indicators of where the true answer is. Rare words like 'tesla' and '1991' carry more useful information in the context paragraphs that they appear in. We used the following formula to calculate IDF.

$$IDF(word) = log \frac{N_{doc}}{N_{word}}$$

So *IDF* for a word captures how rare that this word appears in the whole corpus. The high value of *IDF* indicates lower frequency of the term, and thus higher potential for the term as a named-entity, and thus higher possibility to be part of the answer.

And below is a sample of the *IDF* values:

Table 1: Sample *IDF* for words

tesla	7.52
1949	5.07
university	2.49
in	-1.21

With the IDF generated for the words, we take it as an extra feature for the embedding layer, for the hope that the word with high IDF values can obtain higher "attention" from the model. This is our indirect attempt for named-entity recognition.

The other two additional features are whether or not the word is a key word (*"How", "What", "When", ...*), and if the word appears in the companion context or question.

5 Baseline

There are in total four different layers in the model, Embedding Layer, Attention Layer, Modeling Layer and Prediction Layer.

Embedding Layer: Convert context paragraphs and questions to vector representations with GloVe and encode using GRU.

Attention Layer: Context-to-Question attention is concatenated to the result of embedding layer.

Modeling layer: A fully connected layer followed by a ReLU non-linearity processes the result of Attention Layer.

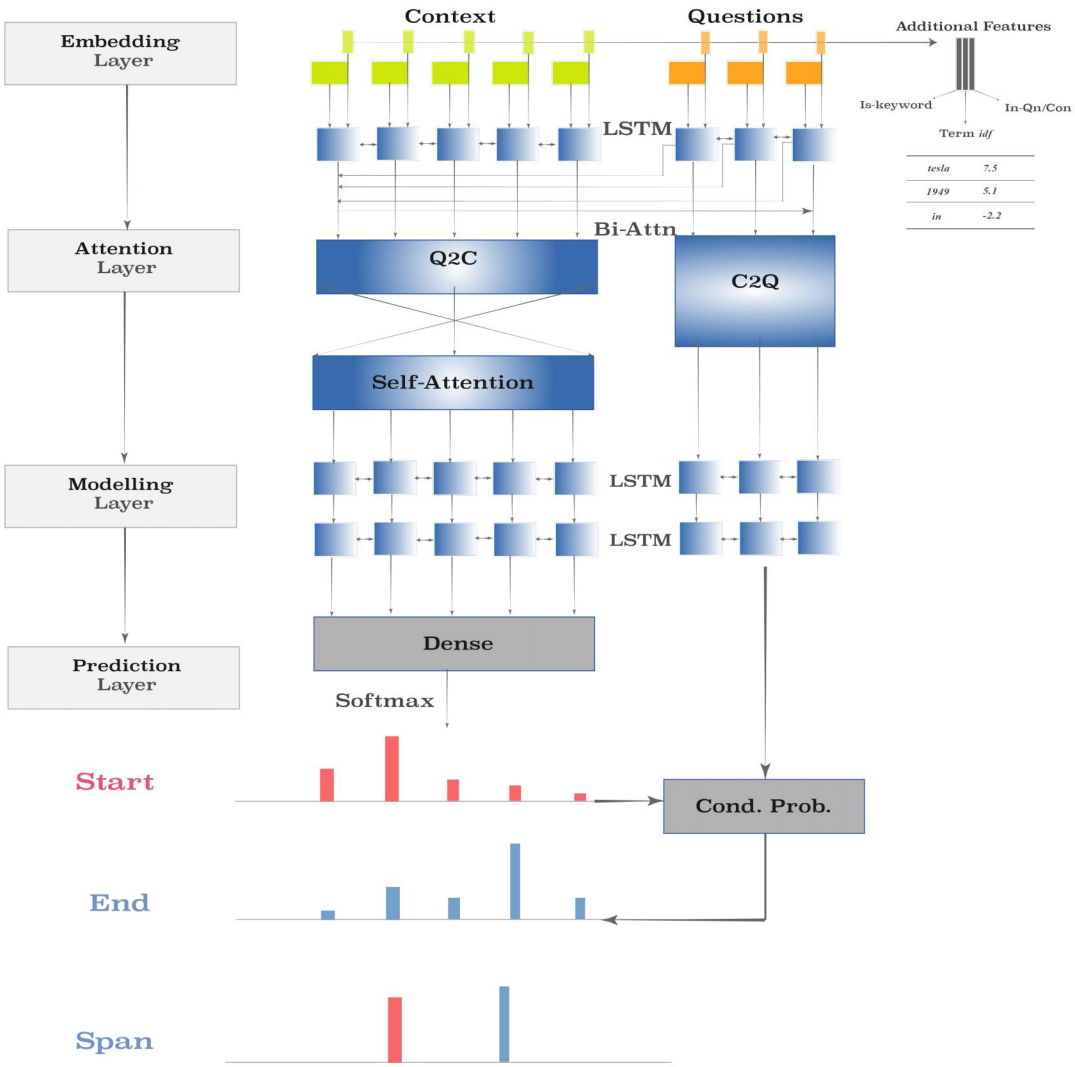
Prediction Layer:

$$Start\_Prob = OutputLayer * W_1$$
$$End\_Prob = OutputLayer * W_2$$

$Start\_Prob$ and End_Prob are of length context_length. The value of $Start\_Prob$ and End_Prob is the probability of answer starting/ending at that index. In other word, the predicted starting index of answer is the index of $max(Start\_Prob)$; The ending index is the index of $max(End\_Prob)$.

6 Our Approach

We encoded GloVe vector representations with LSTM, and then passed the encoded vectors through both self attention layer and bi-directional attention layer and then concatenated the result attentions of these two layers. In the modeling layer we have two layer of LSTM. Please see the following graph for a high level overview.



6.1 Attention Layer

6.1.1 Bi-Directional Attention

Bi-Directional Attention[2] captures both Context-to-Question and Question-to-Context Attention flow. The original paper uses the following equation for the Similarity Matrix:

$$S_{ij} = w_{sim}^T[c_i; q_j; c_i \circ q_i]$$

which requires a concatenation of a large vector, making the model suffering from OOM when batch size is large. Based on our matrix analysis, we instead decompose the w_{sim} vector into three parts w_1, w_2 and w_3, multiply with context C and question Q respectively, and sum with the multiplication of C, Q with w_3 as the diagonalized matrix in between; that is

$$S = C \times w_1 + Q \times w_2 + C \times diag(w_3) \times Q$$

which is more memory-efficient than the original equation and avoids OOM error.

6.1.2 Self Attention

Self attention layer[3] is inserted after the baseline attention layer.

$$e_j^i = V^T \dot{t}anh(W_1 v_j + W_2 v_i)$$

$$\alpha^i = softmax(e^i)$$

$$\alpha^i = \Sigma_{j=1}^N \alpha_j^i v_j$$

$$\{h_1, h_2 ... h_N\} = biLSTM(\{[v_1; \alpha_1], ...[v_N; \alpha_N]\})$$

W_1 and W_2 are weight matrices and V is a weight vector.

6.1.3 Combined Attention

The idea of combined attention is to merge together self attention and bi-directional attention, so that the output of attention layer can include more useful information.

To get combined attention, we first compute self attention and bi-directional attention, then concatenate them and encode them through LSTM.

$$\{h_1, h_2, ...h_N\} = biLSTM(\{[self\_attention_1; bi\_attention_1], ...[self\_attention_N; bi\_attention_N]\})$$

6.2 Modeling Layer

After the attention layer, the modeling layer generates final representation for prediction. We replaced the original dense layer in the baseline with a double-LSTM layer, based on the idea that dense layer is just a linear combination of the attention values, but LSTM can capture the temporal relationship among the attention values.

6.3 Prediction Layer

We improved the prediction layer by linking the process of predicting start of answer and predicting end of answer. That way we can ensure that $start\_index < end\_index$ and $15 > (end\_index - start\_index)$. Based on the baseline prediction layer, we created the following two versions of prediction layer. We trained both versions and used them Ensemble.

Version 1:

$$P_{start}[i] = P_{start}[i] \times \Sigma_{j=i}^{i+15} P_{end}[j]$$

$$P_{end}[i] = P_{end}[i] \times \Sigma_{j=i}^{i-15} P_{start}[j]$$

Version 2:

$$P_{start}[i] = P_{start}b[i] \times max_{j=i}^{i+15} P_{end}[j]$$

$$P_{end}[i] = End\_Prob[i] \times max_{j=i}^{i-15} P_{start}[j]$$

6.4 Ensemble

Our approach for ensemble is to use several different models to sequentially predict the same batch of data, and then take the majority voting of their predictions, as the final prediction. The models we included in our final batch of models are combined attention with prediction layer version 1, combined attention with prediction layer version 2 and two bi-directional attention with prediction layer version 2. The two bi-directional attention models are trained with the same data, but with different random variable initializations.

7 Experiments & Results

Below is the summary of our experiments:

Model	F1	EM
Bi-Directional	0.733	0.627
Self	0.693	0.617
Combined	**0.756**	**0.649**
Ensemble	0.750	0.649

Bi-directional attention model performs better than self attention model. Combined attention performs better than both bi-directional and self attention, which is what we expected, because combined attention has information from both. Ensemble is wore than we expected. One possible reason of the failure of ensemble is that the number of models included in ensemble is too small.

We expect our ensemble to achieve several percent higher accuracy than single model. But it turns out that our ensemble performs slightly worse than the single best model. This may be because that we only used 4 models for the ensemble, 2 of which only have their F1 around 0.73, thus lowering the overall performance. With more time to train several more similar models as the "combined" model, we would expect our ensemble to perform better than single model.

8 Error Analysis

8.1 General Performance

From Figure 3, we see that our model presents two extreme phenomenon: around 60% of the anwers get perfect F1 and EM score, while the other 30% answers get 0 score. This means that our model either fully got the answer or fully missed the answer.

8.2 Performance over Different Answer Length

By comparing the model predictions with the ground truth answers, figure 4 shows that our model presents a larger portion of shorter answers than the ground truth. With this, we analyzed the model performance over different ground-truth answer lengths, as the following two figures show below:

As the two figures shown above, our model generally has a better performance on questions that need shorter answers;

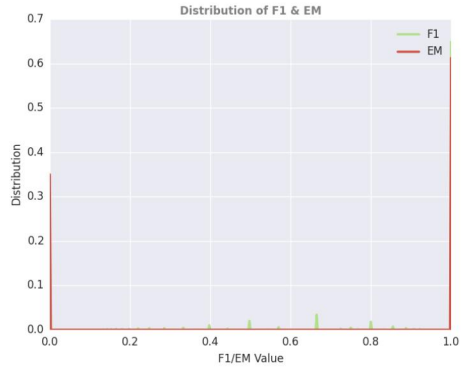


Figure 3: Distribution of F1 & EM

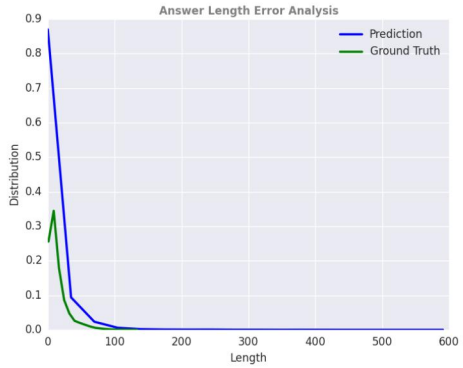


Figure 4: Distribution of Answers by Length

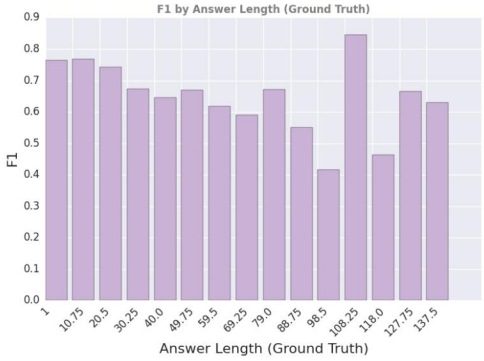


Figure 5: F1 for Different Answer Lengths

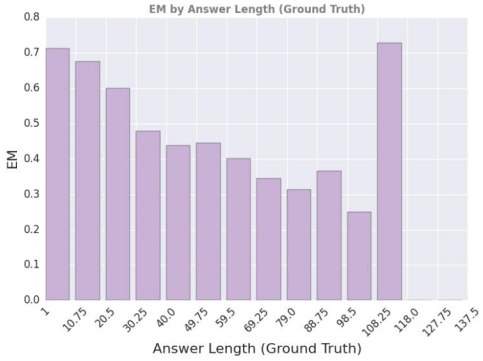


Figure 6: EM for Different Answer Lengths

8.3 Performance for Different Types of Questions

We analyzed our model performance for different questions. As Figure 7 and Figure 8 show below, our model answers the question about *"when"* perfectly, as well as questions for *"where"*, but performed badly on questions about *"what"* and worst on *"how"*. This is reasonable, since *"where"* and *"when"* are both strong indicators for place and time respectively, and they generally need short answers that can be recognized easily. However *"what"* needs a potentially longer answer, that may span multi-words; and *"how"* may need an even more vague answer.

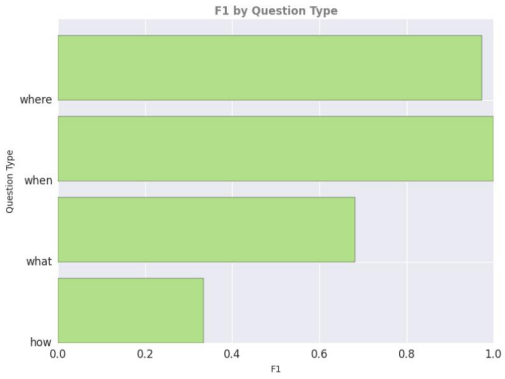


Figure 7: F1 for Question Types

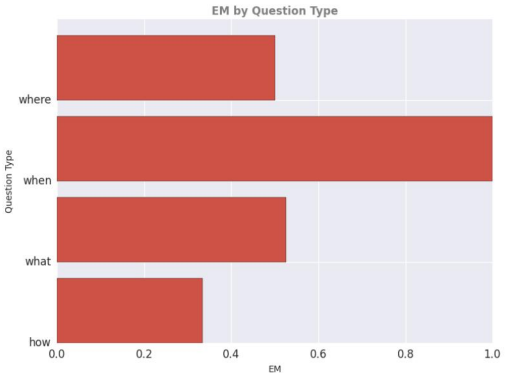


Figure 8: EM for Question Types

One observation is that for the questions of *"where"*, our model has a high accuracy for F1, but low accuracy for EM. This means that the predictions largely overlaps with the ground truth, and probably with several other additional words, or missing some necessary words. This is also reasonable, since a complex phrase for a place can span multiple words with very little correlation among the terms in-between. Without prior knowledge of the places, our model cannot capture the exact place phrase accurately, and may tend to include neighboring unnecessary words as part of the place, or miss several necessary terms for the place phrase.

9 Future Related Work

We can further empower our ensemble method by training more combined attention models. Due to the limitation of time and hardware resources, our current ensemble model only includes two combined attention model, and two other bi-directional models.

To further improve F1 score, the model needs to do better with answering questions that require long answers. Another place where the model should improve is to have a better way to answer questions that start with 'how' and 'what'.

Inspired by the observation that when human answer questions given context, we rely on not just the context paragraph, but also prior knowledge, we propose one potential enhancement to include a knowledge base into the model, similar to the search engine mechanism, where the model can take advantage of prior knowledge for the purpose of question answering.

10 Conclusion

We experimented with Bi-directional Attention, Self Attention and Combined Attention for the question answering model, and achieved solid results. Combined Attention made concrete improvements, compared to Self-Attention and Bi-directional Attention alone. Several different approaches were experimented on the encoding layer, modeling layer, and prediction layers as well.

References

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

[2] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*, 2016.

[3] Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*, 2016.