Character Embedding and Self Attention Mechanism with SQuAD

Stanford CS224N {Default} Project

Name Wei-Hsu Chao
Department of Electrical Engineering
Stanford University
weihsu29@stanford.edu

Name Tsun-Han Huang
Department of Electrical Engineering
Stanford University
tsunhan@stanford.edu

Abstract

The Stanford SQuAD challenge is a reading comprehension contest, which requires us to train a model which can understand the provided contexts and predict the answer correctly. The Stanford SQuAD dataset provides a thorough testing for our model. In this paper, we have explored various ways to achieve better performance to answer the questions from the SQuAD dataset, including BiDAF with character embedding, similarity score based attention layer and self attention mechanism. Our best model achieved the **F1 scores = 64.186** and **EM = 60.524** on the test set.

1 Introduction

Question and Answering is an important topic in Natural Language Processing field. And recently, we care more about reading comprehension style tasks, especially for machine to locate at a specific paragraph of the context when answering the questions (query). Stanford SQuAD challenge provided abundant resources and related data for research groups. In this project, we inspect the implementation of BiDAF paper[1]. We first explore what the impacts of character embedding on the model performance. BiDAF model already has bidirectional Context2Query and Query2Context attention mechanism in their implementation. So, we try to add additional self-attention mechanism to the BiDAF model to see if self-attention can help improve the BiDAF model performance or not, and then discuss the training and testing results in our experiments.

2 Related work

In this project, we explored various implementations for the SQuAD challenges. Here are what we have studied and explored:

- **Character-level Embeddings[1]:**
  The first step we do is to further improve the baseline model. In addition to the word-level embedding, we add additional character-level embedding within the embedding layer to make our model back to the original BiDAF model[1]. This enables the model to process the words in the character level, so that we can better represent the unknown words.

- **Similarity Score Based Attention Layer[1]:**
  The paper proposed the way to calculate the similarity score of two vectors first, then use the similarity scores to calculate the attention vectors. Besides Context2Query and Query2Context attention in the BiDAF model, this paper inspires us to implement an additional self-attention mechanism by using the similarity scores between context words.

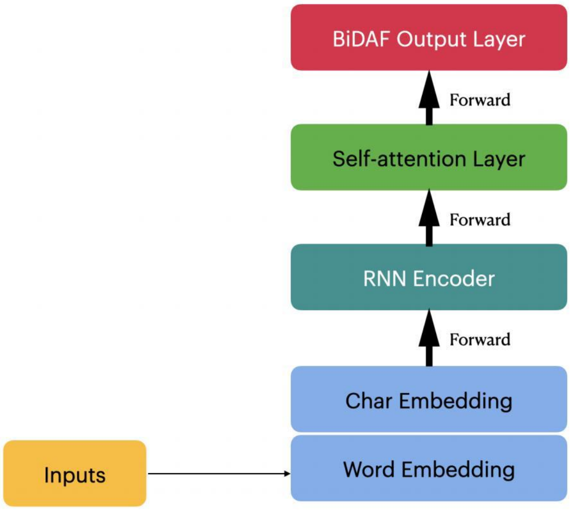
- **Context2Context Self Attention[4]:**
  This paper proposed to use self-matching (self-attention) mechanism in the SQuAD Chal-

lenge and they also reached the best leaderboard score at that time. They used additive attention mechanism, and we choose to use multiplicative attention and add it to BiDAF model to see how it compares to the results of obtaining self-attention vectors from similarity scores.

Based on the papers and the baseline model, we added additional character embedding, and we added self-attention mechanism onto the BiDAF model with two different self-attention implementations. Then we discuss the experiment results and analyze the impacts of self attention on BiDAF model's performance.

3 Approach

Figure 1: Model Architecture



- Main Approach

1. Character Embedding

We added a character embedding back into the original BiDAF without Character Embedding baseline model. We loaded the provided character embedding vectors file, and let the model keep updating the character embeddings during the training process. The dimension of the provided character embedding vectors is 64. And we used feed-forward neural network to projected the vectors into shorter vectors with dimension 8. Then we applied Convolution Neural Network (CNN) with kernel size (8, 5) on the 2-D vectors with (height, width) = (8, maxLengthOfCharsInAWord), so it could generate a 1-D vector with length maxLengthOfCharsInAWord. And we perform 1-D max pooling on it. After applying CNN and max pooling, the character embeddings can be concatenated with the word embedding and fed into LSTM encoders. And the remaining model structure is the same as the baseline model.

2. Self Attention Based on Modified Similarity Scores

In the original BiDAF model[1], it has Bidirectional Attention Mechanism after Word Embed Layer and and Contextual Embed Layer, which is Context2Query Attention and Query2Context Attention. The BiDAF paper[1] proposed a way to calculate the similarity score between a context vector and a query vector, each vector representing a context word or a query word. So, they would generate a similarity matrix with number of rows equaling to number of query words and number of columns equaling to number of context words. Using softmax on each row can get the attention scores of context words when we look at a query word. Using softmax on each column can get the attention scores of query words when we look at a context word.

In our approach, in addition to Context2Query and Query2Context, we also added additional Self Attention Mechanism, Context2Context Attention. We generated another similarity matrix with number of rows and number of columns both equaling to number of context words, basing on the modified similarity score calculating function in the BiDAF implementation we referenced[2]. So, taking softmax on each column can get the attention scores of context words. We would like to experiment whether adding Context2Context attention into the original BiDAF model can improve the model performance or not.

3. Self Attention Based on Multiplicative Attention

In addition to self attention based on similarity scores, we also experimented on multiplicative attention: $\mathbf{e}_{t,i} = \mathbf{c}_t^T \mathbf{W} \mathbf{c}_i$, which is revised from the additive attention $\mathbf{e}_{t,i} = \mathbf{v}^T tanh(\mathbf{W}_1 \mathbf{c}_i + \mathbf{W}_2 \mathbf{c}_t)$ used in paper [4]. We choose to implement multiplicative attention because it is more trainable than additive attention with the time and GPU memory limits of this project. We added multiplicative Context2Context attention to BiDAF model[1] to see how multiplicative self attention compares to modified similarity scores based self attention mentioned in Main Approach 2.

- Baseline Model

  The baseline model is provided by Stanford CS224N in the default project starter code. It is revised from the BiDAF paper[1]. And the original BiDAF paper[1] described the model details. The BiDAF model includes character embeddings to be more representative about unknown tokens. And the baseline model removed the character embedding part to make the model training process more memory and time efficient.

- Implementation

  We provided the Github Links to the baseline starter code[3] and the BiDAF model implementation[2] we have referred to and revised from.

- Add Up from Our Team

  Since the reference implementation[2] used character embedding with dimension 8, so we added a feed-forward neural network in our model to project the provided 64-dimension character embeddings into dimension 8.

  We also added Context2Context Attention based on the modified similarity score calculating method in the reference implementation[2] to experiment on the effects of adding self attention mechanism into the original BiDAF model[1].

  Finally, we added multiplicative Context2Context attention to the BiDAF model[1] to compare it with similarity-score-based Context2Context attention.

4 Experiments

- **Data**: The dataset we use are mainly from the official SQuAD 2.0 dataset with modifications by CS224n staff for course purpose. The details are as follow,
  - Train (129,941 examples): All taken from the official SQuAD 2.0 training set.
  - Dev (6078 examples): half of the official dev set, randomly selected.
  - Test (5915 examples): The remaining examples from the official dev set, plus hand-labeled examples

- **Evaluation method**: There are two evaluation metrics we use for our models.
  - **F1 Scores:**

$$F1 = \frac{2 \times precision \times recall}{(precision + recall)}$$

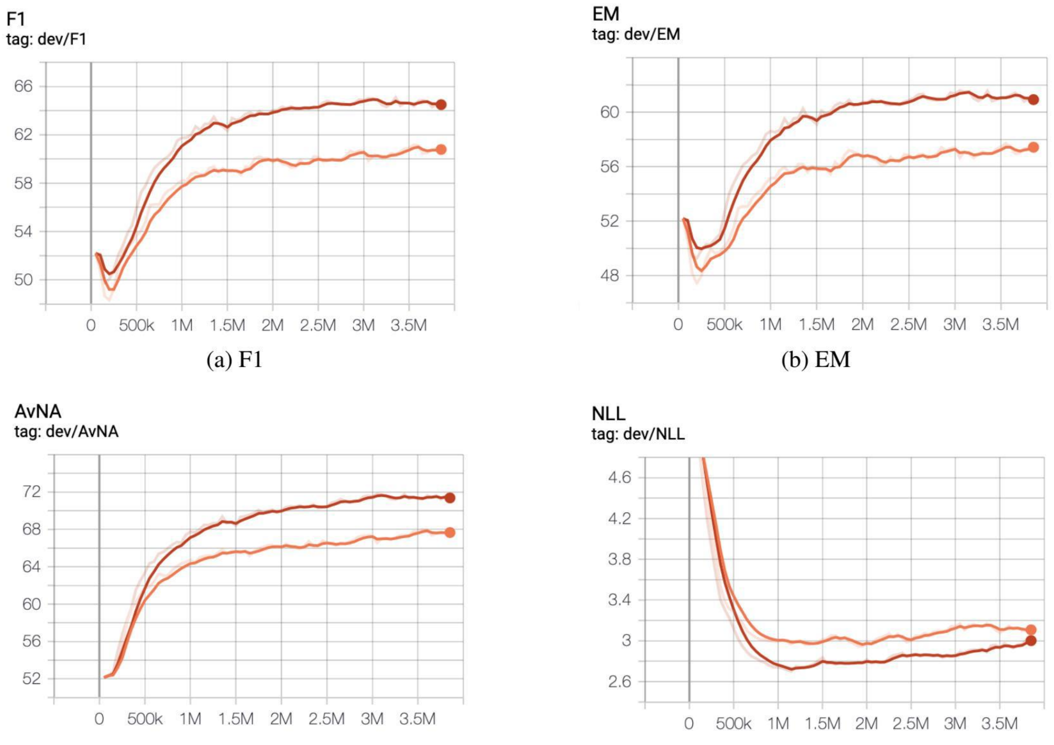
  - **EM (Exact Match) Scores:** EM is a binary measure (i.e. true/false) of whether the system output matches the ground truth answer exactly.

- **Experimental details**: We have trained 4 different models so far, including baseline model, BiDAF model with character embedding, attention layer with modified similarity score model and self-attention model. Here are the hyperparameters we used:
  - Learning rate = **0.5**
  - Weight decay = **L2 weight decay**
  - epochs = **30**
  - dropout prob = **0.2**
  - optimizer = **Adadelta**

  Training time:
  - Baseline model: **18 hours**
  - character embedding model: **31 hours**
  - Attention layer with modified similarity scores: **30 hours**
  - Self-attention: **30 hours**

- **Results**:

Figure 2: Comparison of baseline and character embedding model



(a) F1

(b) EM

(c) AvNA

(d) NLL

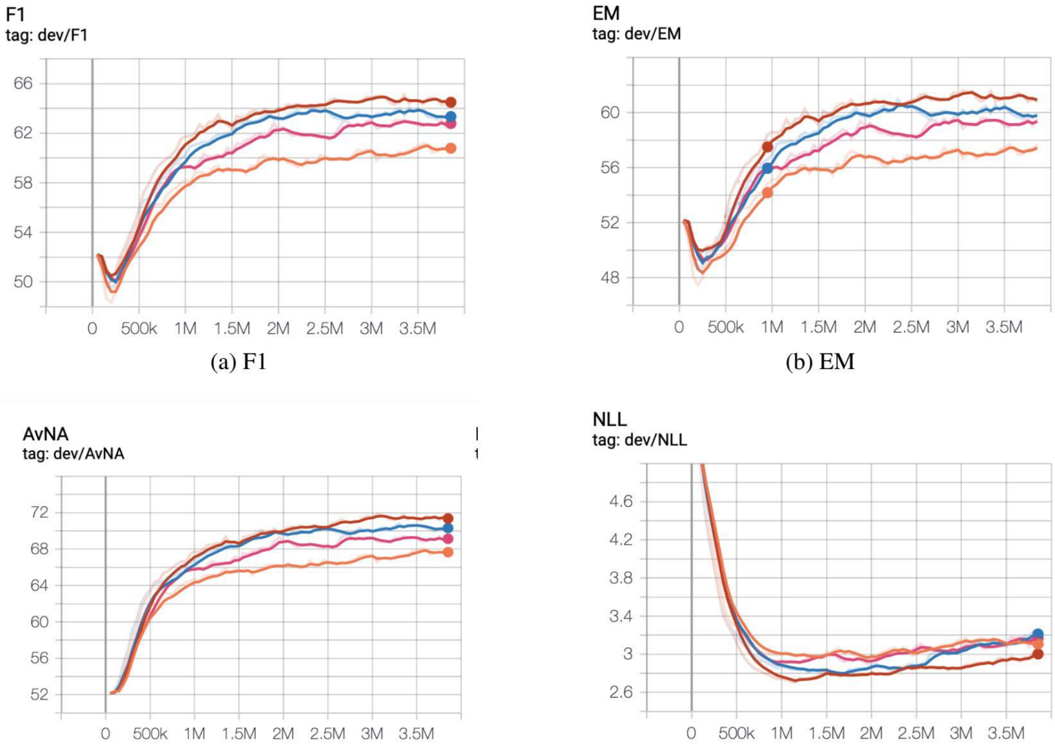
orange line: baseline model
red line: character embedding model

- **Baseline model** - F1 scores = **60.86**, EM = **57.40**, AvNA = **67.69**, NLL = **3.107**
  (Results from validation leaderboard: F1 = **61.17**, EM = **57.70**.)

- **BiDAF with character embedding model** - F1 scores = **64.40**, EM = **60.76** , AvNA = **71.25**, NLL = **3.036**
  (Results from validation leaderboard: F1 = **65.099**, EM = **61.553**, test leaderboard: F1 = **64.186**, EM = **60.524**)

  As the figure shows, both F1 and EM scores have some increase after adding character embedding. Further, the AvNA scores also gain improvement. Regarding the NLL loss, we can see that they both went into plateau after 2M. With character embedding, the model becomes more representative for words, especially for the unknown words, and has improvements in the F1 and EM scores.

Figure 3: Comparison of different models



(a) F1

(b) EM

(c) AvNA

(d) NLL

orange line: baseline model
red line: character embedding model
blue line: similarity score based self attention model
pink line: multiplicative self attention model

- **Similarity score based self attention model** - F1 scores = **63.41**, EM = **59.89**, AvNA = **70.44**, NLL = **3.235**
  (Results from validation leaderboard: F1 = **64.153**, EM = **60.948**, test leaderboard: F1 = **62.652**, EM = **58.850**)

- **Multiplicative self attention model** - F1 scores = **62.91**, EM = **59.54**, AvNA = **69.25**, NLL = **3.195**
  (Results from validation leaderboard: F1 = **63.087**, EM = **59.452**, test leaderboard: F1 =

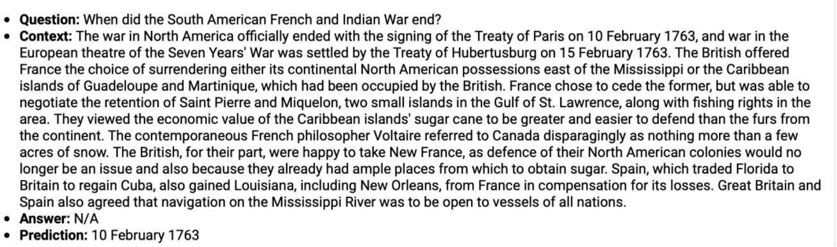
63.053, EM = **59.256**)

As the figure shows, adding Context2Context attention to BiDAF model with character embedding can not improve the F1 and EM scores. And the self attention based on modified similarity score has slightly better F1 and EM scores than multiplicative self attention.

Our experiment results show that adding self attention will not necessarily improve the model performance. One of the possible reasons is that attending to additional context words may distract model's attention and introduce some interference. The original BiDAF model has performed really well with Context2Query and Query2Context attention. Adding additional Context2Context may make the model unnecessarily complicated and can not generalize well on the validation and testing dataset. And there are more parameters in multiplicative attention (hiddenSize * hiddenSize) than similarity score based attention (3 * hiddenSize), so the performance slightly decreased with more parameters included and more complex model.

5 Analysis

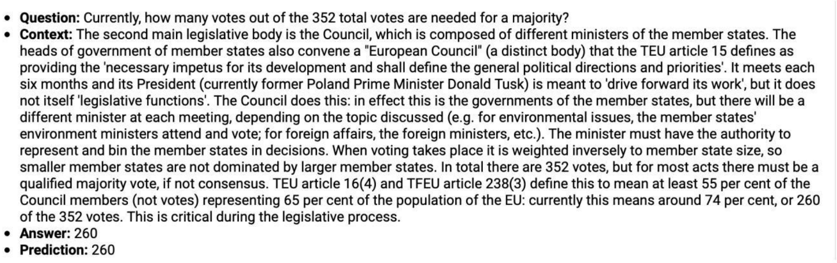
Figure 4: Modified similarity score model paragraph comprehension



- **Question:** When did the South American French and Indian War end?
- **Context:** The war in North America officially ended with the signing of the Treaty of Paris on 10 February 1763, and war in the European theatre of the Seven Years' War was settled by the Treaty of Hubertusburg on 15 February 1763. The British offered France the choice of surrendering either its continental North American possessions east of the Mississippi or the Caribbean islands of Guadeloupe and Martinique, which had been occupied by the British. France chose to cede the former, but was able to negotiate the retention of Saint Pierre and Miquelon, two small islands in the Gulf of St. Lawrence, along with fishing rights in the area. They viewed the economic value of the Caribbean islands' sugar cane to be greater and easier to defend than the furs from the continent. The contemporaneous French philosopher Voltaire referred to Canada disparagingly as nothing more than a few acres of snow. The British, for their part, were happy to take New France, as defence of their North American colonies would no longer be an issue and also because they already had ample places from which to obtain sugar. Spain, which traded Florida to Britain to regain Cuba, also gained Louisiana, including New Orleans, from France in compensation for its losses. Great Britain and Spain also agreed that navigation on the Mississippi River was to be open to vessels of all nations.
- **Answer:** N/A
- **Prediction:** 10 February 1763

- In this questions, the answer should be "N/A" while the baseline model wrongly predicts the only date in the question. This means that the baseline model doesn't completely understand the given paragraph. On the other hand, with our modified similarity score attention model and self-attention model, we correctly predict the answer to "N/A".

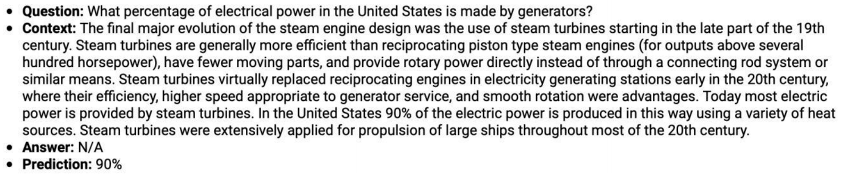
Figure 5: Quantity prediction



- **Question:** Currently, how many votes out of the 352 total votes are needed for a majority?
- **Context:** The second main legislative body is the Council, which is composed of different ministers of the member states. The heads of government of member states also convene a "European Council" (a distinct body) that the TEU article 15 defines as providing the 'necessary impetus for its development and shall define the general political directions and priorities'. It meets each six months and its President (currently former Poland Prime Minister Donald Tusk) is meant to 'drive forward its work', but it does not itself 'legislative functions'. The Council does this: in effect this is the governments of the member states, but there will be a different minister at each meeting, depending on the topic discussed (e.g. for environmental issues, the member states' environment ministers attend and vote; for foreign affairs, the foreign ministers, etc.). The minister must have the authority to represent and bin the member states in decisions. When voting takes place it is weighted inversely to member state size, so smaller member states are not dominated by larger member states. In total there are 352 votes, but for most acts there must be a qualified majority vote, if not consensus. TEU article 16(4) and TFEU article 238(3) define this to mean at least 55 per cent of the Council members (not votes) representing 65 per cent of the population of the EU: currently this means around 74 per cent, or 260 of the 352 votes. This is critical during the legislative process.
- **Answer:** 260
- **Prediction:** 260

- In this question, both baseline model and character embedding model correctly answer this questions, but the model with modified similarity scores predicts the answer of "352" instead of "260". This means that when it comes to answering quantity questions, the first two models have better performance.

Figure 6: Self-attention comprehension prediction



- **Question:** What percentage of electrical power in the United States is made by generators?
- **Context:** The final major evolution of the steam engine design was the use of steam turbines starting in the late part of the 19th century. Steam turbines are generally more efficient than reciprocating piston type steam engines (for outputs above several hundred horsepower), have fewer moving parts, and provide rotary power directly instead of through a connecting rod system or similar means. Steam turbines virtually replaced reciprocating engines in electricity generating stations early in the 20th century, where their efficiency, higher speed appropriate to generator service, and smooth rotation were advantages. Today most electric power is provided by steam turbines. In the United States 90% of the electric power is produced in this way using a variety of heat sources. Steam turbines were extensively applied for propulsion of large ships throughout most of the 20th century.
- **Answer:** N/A
- **Prediction:** 90%

- In this question, the character embedding model wrongly predict the answer to be "90%", which is irrelevant to the question. However, the self-attention model predicts the answer correctly to "N/A. This means that it can understand the context of this question while the character embedding model cannot.

6 Conclusion

In this project, we have demonstrated the effectiveness of character embedding. According to our experiment results, adding Context2Context self attention mechanism can not improve the performance of the BiDAF model. The BiDAF model with character embedding performs well with its Context2Query attention and Query2context attention. Adding self attention to this model will include additional interference when the context words attend not only to the query words, but the context words itself, which slightly reduced the model performance. For the future work, we can add additive attention to the BiDAF model to see how it compares to the two attention implementations we use. In addition, there are plenty of modern techniques, including Transformer and Reformer, can be further explored to find the best performing model on SQuAD challenge.

References

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

[2] Github Links of BiDAF Implementation : https://github.com/GauthierDmn/question_answering

[3] Github Links of Baseline Starter Code : https://github.com/minggg/squad

[4] Natural Language Computing Group, Microsoft Research Asia. R-Net: Machine Reading Comprehension with Self-Matching Networks. 2017