SQuAD - Refined Implementation of Contextually Enriching Passage Sequences (SQuAD-RICEPS)

Stanford CS224N Default Project

Daniel Classon
Department of Computer Science
Stanford University
dclasson@stanford.edu

Thomas Jiang
Department of Computer Science
Stanford University
twjiang@stanford.edu

Ryan Peng
Department of Computer Science
Stanford University
pengryan@stanford.edu

Abstract

We evaluated the task of machine reading and question answering on the Stanford Question Answering Dataset (SQuAD) 2.0 using an improved version of Bidirectional Attention Flow (BiDAF) model. We combined the BiDAF-No-Answer model with an added character embedding layer, trilinear attention layer, and self-attention layer described in Clark et. al (2017). Our goal was to score high on SQuAD 2.0 according to Exact Match (EM) and F1 scores. The final EM and F1 score of the model on the test set were 59.510 and 62.708, respectively. This is an improvement from the baseline's EM score of 56.3 and F1 score of 59.4.

1 Key Information to include

- Mentor: Nope
- External Collaborators (if you have any): Nada
- Sharing project: Nah
- Using shared late days: Yes, using 1 shared day from our 1.67 (4 + 0 + 1) and 1 penalized late day

2 Introduction

Natural Language Processing (NLP) is defined as a branch of artificial intelligence that focuses on the interactions between humans and computers using the natural language. The study of NLP has been around for decades and developed out of the field of linguistics with the increased rise of computers and technology. Question Answering (QA) represents one of the core fundamental areas of research in NLP. In QA tasks, a model is given a paragraph of text (context) and is then asked a question about the text. An ideal model would provide the correct answer, as predetermined by a human. One of the most popular datasets for benchmarking and analyzing this task is the Stanford Question Answering Dataset (SQuAD). Current approaches to QA tasks still contain problems including the possibility of the system focusing on irrelevant portions of the context passage as well as understanding that not all questions have a valid answer given in the specified context. This project aims to explore combined layers of attention and improve upon a baseline approach under EM and F1 scores on SQuAD 2.0. We present SQuAD-RICEPS, which uses a character embedding layer, trilinear attention layer, and self-attention layer to improve performance and accuracy over the baseline model.

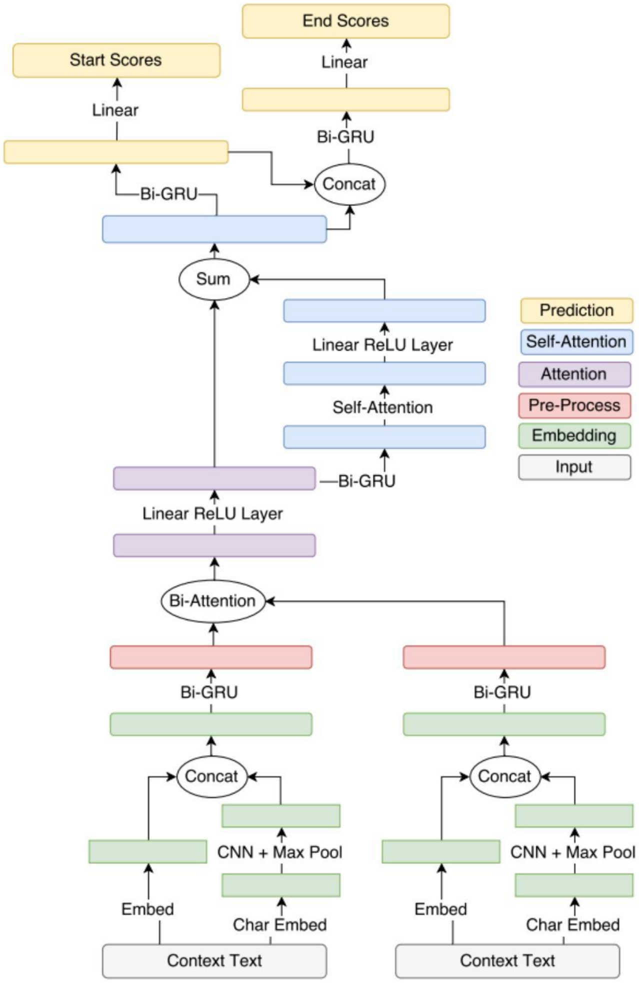


Figure 1: Model from Clark et. al (2017)

3 Related Work

Microsoft (2017) aims to build an end-to-end neural networks model for reading comprehension style question answering, which aims to answer questions from a given passage. This research was done by the Natural Language Computing Group, Microsoft Research Asia and mainly focused on the Stanford Question Answering Dataset (SQuAD) and the Microsoft Machine Reading Comprehension (MS-MARCO) dataset. The built R-NET model consists of four parts: the recurrent network encoder to build representation for questions and passages separately, the gated matching layer to match the question and passage, the self-matching layer to aggregate information from the whole passage, and lastly, the pointeR-NETwork based answer boundary prediction layer. This model consists of new ideas including a gated attention-based recurrent network and a self-matching mechanism. This model achieves a high accuracy of 72.3 percent (exact match) and its ensemble model increases the accuracy to 76.9 percent. At the time, this model held first place on the SQuAD leaderboard and further aims to improve question-answering machine reading comprehension and introduces new networks that improve machine learning NLP methods.

Microsoft (2017) introduces two new network layers to machine comprehension tasks:

1. Gated Attention-Based Recurrent Network: This layer is needed to incorporate question information into passage representation. This networks stems from attention-based recurrent networks and adds an additional gate to determine the importance of information in the passage regarding a question. This particular implementation is different from the gates in an LSTM or GRU

$$att(Q,K,V) = softmax(\frac{QK^T}{d_k})V$$

Figure 2: Scaled dot product self-attention for multiheaded self-attention

$$f(q,c) = W_0[q,c,q \odot c]$$

Figure 3: Trilinear similarity function used in trilinear attention, Yu et. al (2018)

architecture in that the additional gate is based on the current passage word and its attention-pooling vector of the question, which focuses on the relation between the question and current passage word. The gate effectively model the phenomenon that only parts of the passage are relevant to the question in reading comprehension and question answering.

2. Self-Matching Attention: One problem with gated attention-based recurrent networks is that the generated question-aware passage representation is limited in knowledge of context. The paper aims to solve this problem by directly matching the question-aware passage representation against itself. In doing so, it dynamically collects evidence from the whole passage for words in passage and encodes the evidence relevant to the current passage word and its matching question information into passage representation. Another gate is added to adaptively control the input of the RNN. This method extracts evidence from the whole passage based on the question and needed information. This method was also mentioned and utilized in Clark et al. (2017).

Clark et al. (2017) considers the problem of adapting neural paragraph-level question answering models to the case where entire documents are given as context input. The proposed solution trains models to produce well calibrated confidence scores for their results on individual paragraphs. Clark et al. (2017) samples multiple paragraphs from the documents during training, and uses a shared normalization training objective that encourages the model to produce globally correct outputs. Experiments from Clark et al. (2017) demonstrate strong performance on several document QA datasets. Overall, the team was able to achieve an F1 score of 71.3 on the web portion of TriviaQA.

4 Approach

Our approach was to implement the BiDAF-No-Answer model using the self-attention layer described in Clark et. al (Figure 1), which we refer to as SQuAD-RICEPS in our experimental results. We then extended SQuAD-RICEPS to use multiple self-attention layers and character embeddings.

We based off our embedding and encoding layers with the given BiDAF code. We enhanced it by implementing and experimenting with character embeddings. As with Figure 1, we put our character embeddings through a 1d convolution network and then ran maxpool. We then concatenated this result with the word embedding and put it through the Highway layer before returning it. We thought that using this approach we could improve the accuracy of our training model since it takes into account the semantic meanings of a group of characters.

For our newly modified attention, we based off our model from the attention layout given in Figure 1 and combined it with the default BiDAF that we were given for the baseline. We first passed the result of our BiDAF through a reLU linear layer. Next we branched off by creating a layer of residual attention which was passed through a bi-directional GRU. We then apply the self attention mechanism from the passage to itself. Another approach that we added was to further pass that attention into a multiheaded attention layer. As with the baseline attention, we concatenated our results and then passed them through a final linear ReLU layer as seen in Figure 1. Lastly we summed our residual layer and input layer with ReLU activation.

For the self-attention mechanism, we experimented with multiheaded scaled dot product attention (Figure 2) and trilinear attention as described in Yu et. al (2018) (Figure 3). The multiheaded scaled dot product attention was taken from Assignment 5. We used existing implementations of the

TriLinearAttention and TimeDistributed functions that existed in the AllenAI NLP library, maintained by one of the co-authors of Clark et. al (2017).

5 Experiments

- **Data**: We are using the SQuAD 2.0 dataset with the given pre-processing as described in the default handout. This dataset contains 100,000+ question-answer pairs posed by crowd-workers on a set containing 500+ wikipedia articles. It also contains 50,000+ unanswerable questions.
- **Evaluation method**: Our model's performance will be evaluated using the Exact Match (EM) and F1 scoremetrics described in the default project handout. We will be comparing these scores with the scores given in our baseline model. We have three existing scores that we can compare against: R-NET single model – 72.3 EM (SQuAD 1.0), R-NET ensemble model – 76.9 EM (SQuAD 1.0), BiDAF-No-Answer single model – 59.174 EM and 62.093 F1. These will give a good indication as to how well our model is performing in reference to defined and highly optimized models.
- **Experimental details**: We tried many model configurations, the first being the addition of multiheaded self-attention to the baseline BiDAF model. After the results performed worse than the baseline, we changed our model of self-attention to TriLinear attention and optimized our self-attention mechanism according to Figure 1–this is shown as "SQuAD-RICEPS (pink)" in the above tables. From here we experimented with lowering the learning rate from 0.5 to 0.1 (in green in Figure 2) as well as adding in multiheaded self-attention as a third attention layer (after BiDAF and TriLinear attention). Then, we implemented character embeddings. To improve training time, we also increased the learning rate from 0.5 to 0.8 when training character embeddings.
- **Results**: Our EM, F1, and NLL results from Tensorboard are in Figures 4-6. MHSDPA stands for an additional Multihead Scaled Dot Product Attention layer, CE stands for Character embeddings, and lr stands for learning rate (mentioned if not the default value of 0.5).

Model Configuration	EM	F1	NLL	Train time
BiDAF w/ MHSDPA (orange)	50.56	52.4	3.73	10h
SQuAD-RICEPS w/ lr 0.1 (green)	55.8	58.8	2.78	13h
BiDAF Baseline (orange)	56.5	59.6	2.98	5h
SQUAD-RICEPS w/ CE (pink)	57.1	60.0	2.82	15h
SQuAD-RICEPS (pink)	59.5	62.4	2.69	14h
SQUAD-RICEPS w/ CE, MHSDPA, lr 0.8 (gray)	60.0	63.0	2.71	21h
SQuAD-RICEPS w/ MHSDPA (blue)	**60.6**	**63.4**	**2.69**	21h
SQUAD-RICEPS w/ CE, MHSDPA, lr 0.8, Test Set (N/A)	56.7	59.6	N/A	N/A
SQuAD-RICEPS w/ MHSDPA, Test Set (N/A)	59.5	62.7	N/A	N/A

Table 1: Experimental Performance on Dev Set

6 Analysis

We are pleased that our final triple-layered attention model outperforms our vanilla SQuAD-RICEPS submission for the milestone as well as the BiDAF baseline. We did try fine-tuning the learning rate–the green line from Figure 4 shows the progression of a 0.1 learning rate. In our case, a higher learning rate proved to achieve a better EM and F1 score. We can also see that the time to train the models increases as the number of layers increases.

Our last model configuration, which combined the triple-layered attention model with character embeddings and a learning rate of 0.8, had comparable performance on the dev set. The addition of character embeddings would theoretically boost our test set scores by allowing our model to answer questions with unseen words. However, this character embedding model's performance on the test set was much worse than expected–while only 0.4 points lower on the dev set, it scored an average of 3.2 points lower on the test set when compared to our final model. We theorize that the model's

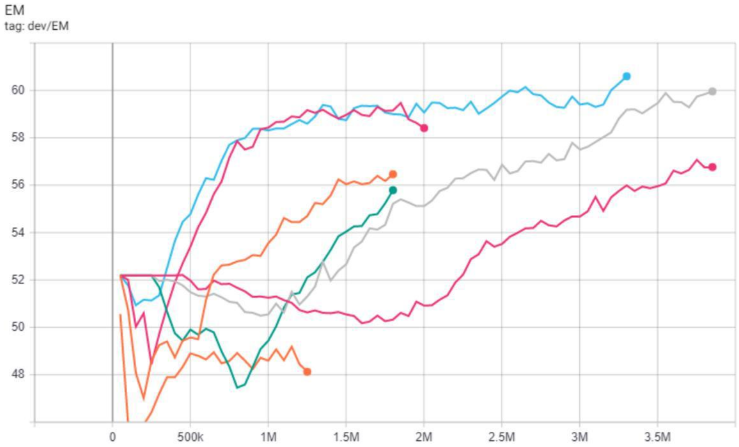


Figure 4: Dev EM / Training Steps

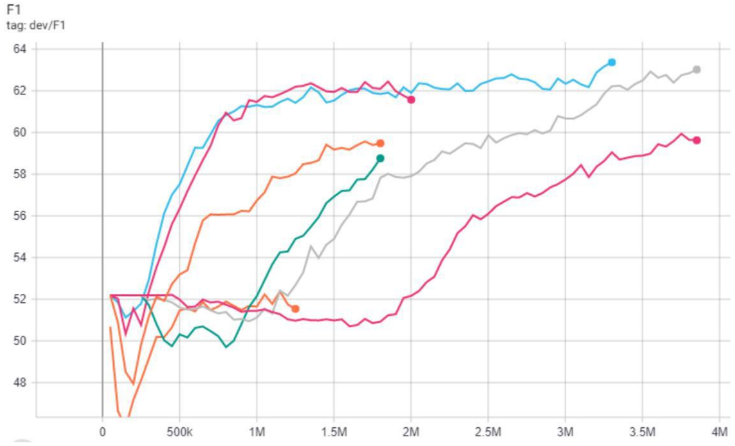


Figure 5: Dev F1 / Training Steps

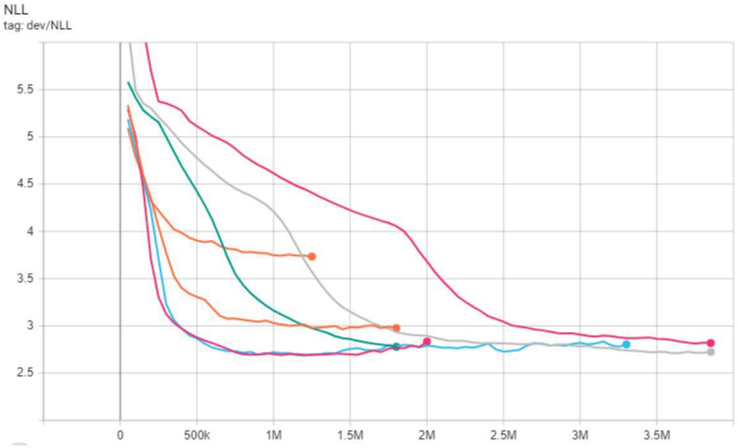
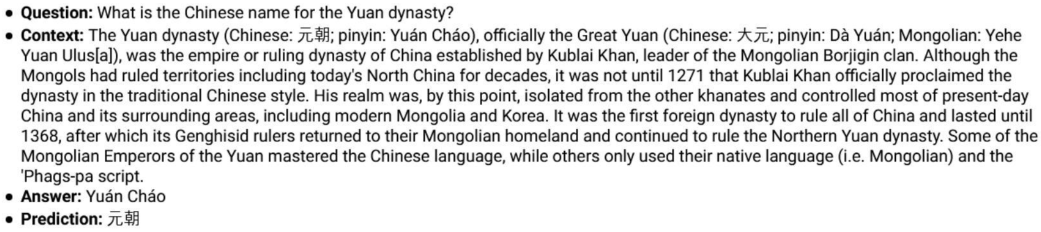


Figure 6: Dev NLL / Training Steps



- **Question:** What is the Chinese name for the Yuan dynasty?
- **Context:** The Yuan dynasty (Chinese: 元朝; pinyin: Yuán Cháo), officially the Great Yuan (Chinese: 大元; pinyin: Dà Yuán; Mongolian: Yehe Yuan Ulus[a]), was the empire or ruling dynasty of China established by Kublai Khan, leader of the Mongolian Borjigin clan. Although the Mongols had ruled territories including today's North China for decades, it was not until 1271 that Kublai Khan officially proclaimed the dynasty in the traditional Chinese style. His realm was, by this point, isolated from the other khanates and controlled most of present-day China and its surrounding areas, including modern Mongolia and Korea. It was the first foreign dynasty to rule all of China and lasted until 1368, after which its Genghisid rulers returned to their Mongolian homeland and continued to rule the Northern Yuan dynasty. Some of the Mongolian Emperors of the Yuan mastered the Chinese language, while others only used their native language (i.e. Mongolian) and the 'Phags-pa script.
- **Answer:** Yuán Cháo
- **Prediction:** 元朝

Figure 7: A prediction by our final model

weakness was its training time and learning rate, which should have been fine tuned more to allow the scores to reach their maximum. We also noticed that the NLL curves for both of our character embedding models (in pink and gray in Figure 6) looked more like piecewise functions rather than logarithmic. We are not sure why the learning curve is so disjointed but theorize that it has to do with the time required to train the embeddings before minimizing the objective.

There are actually a few cases where our model got the answer "wrong" but the predicted result seems correct. For example, in Figure 7, our model predicted with the Chinese characters for Yuan Chao rather than the pinyin. Based on the scoring rules for EM, it seems like our model would be penalized for not predicting the gold answer or a substring of it. We theorize that most of these cases are due to the inconsistent quality of the gold answers in the dataset rather than a trait of our model.

7 Conclusion

In this project, we explored a variety of methods for QA on SQuAD 2.0 including TriLinear Attention, Multi-headed Dot Product Self Attention and character embeddings. Our experiements show that the best result for the QA task was evaluated on the dev set with an EM score of 60.6 and F1 score of 63.4. However, the test set evaluation was only able to achieve an EM score of 59.5 and F1 score of 62.7. Furthermore, our experiments shows that adding extra layers may not increase EM and F1 performance and makes the model more complex and increases time to train. Given more time, we would like to fine-tune the hyper-parameters for SQuAD-RICEPS w/ MHSDPA which can lead more better results on the test set and evaluate the performance of other attention layers.

References

[1] Clark et. al. Simple and Effective Multi-Paragraph Reading Comprehension. 2017.

[2] Microsoft Research Asia Natural Language Computing Group. R-NET: Machine Reading Comprehension With Self-Matching Networks. 2017.

[3] Yu et. al. QANet: Combining Local Convolution with Global Self-attention for Reading Comprehension. 2018.