RE-Net: A Character-Based Model Replicating R-Net

Arvind Subramanian
Undeclared
Stanford University
arvindvs@stanford.edu

Aditya Iswara
Department of Computer Science
Stanford University
iswara@cs.stanford.edu

Abstract

Contextual question answering is a problem that has garnered a lot of attention in the past few years, especially in the field of Natural Language Processing using deep learning techniques. The development of a model that can extract salient answers to a question from a large block of context text would be extremely valuable in streamlined information extraction. These kinds of question-answering models would be particularly useful in texts like legal documents or textbooks. The SQuAD Challenge is a relatively recent question-answering task that incentivizes the development of a model that approaches human-level performance. We propose an initial improvement on a baseline BiDAF model by adding a character-level embedding to the already existing word-level embedding. Additionally, we aim to use the self-matching attention and gated attention network incorporated by R-Net, which has been shown to be more effective than the standard BIDAF Model. Our contributions to this paper will be the combination of character-level embeddings with self-matching attention layers similar to that of R-Net in order to validate existing models on the SQuAD v2.0 dataset.

1 Introduction

The development and use of contextual question answering models has grown rapidly in the last few years, which has applications in a variety of contexts. One dataset that has grown in popularity within the research community has been the Stanford Question Answer Dataset (SQuAD). SQuAD is a curated dataset of questions and answers in which a context paragraph is provided and in which the model is meant to extract a span of the passage to answer a question.

One important thing to study regarding this dataset is the differences in the versions that have been released. The SQuAD1.1 dataset only evaluated whether a model was able to extract a salient span that answered the question presented, regardless of whether the question had an answer in the context or not. The SQuAD2.0 dataset seeks to account for this by testing whether models can not only find the span within the passage, but determine if the question has an answer in the context at all.

In this research, we will be constructing models based on the SQuAD2.0 dataset, as this is where most current literature is focused on. This dataset is more comprehensive and provides a robust question-answering model that is able to predict when it cannot provide an answer.

This question-answering problem is important for a few reasons. The goal of these model constructions is to outperform a human in extracting a relevant span of the context passage. A model that could extract information as well or better than a human, but exponentially faster, would be extremely useful in multiple contexts. Rather than searching for the answer to a question in a history textbook manually, a student could simply ask the model a question and receive a mostly correct response almost instantaneously. Rather than searching an entire corpus of research papers for information on a research technique, a researcher could simply ask the model a question and receive this information. The applications of this technology are boundless.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

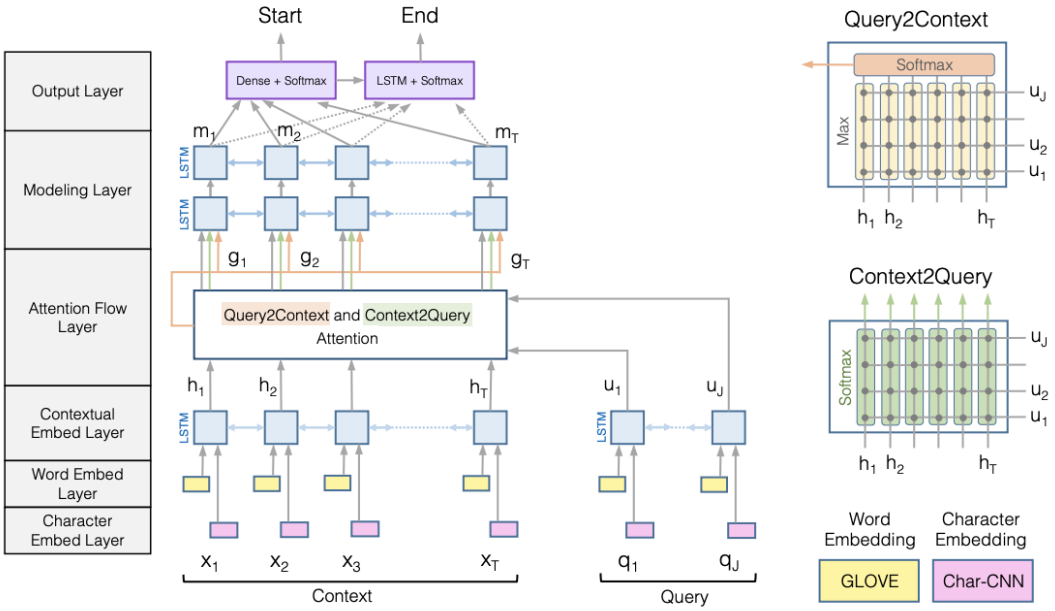

Figure 1: Baseline Model

On this note, it is prudent to mention that models have already been constructed on the SQuAD1.1 dataset that outperform humans in extracting information. The top models on this earlier dataset include *BERT (ensemble) by Google AI Language Group*[1] and *nlent (ensemble) by the Microsoft Research Asia Group*[2].

2 Related Work

The baseline model for this project is shown in Figure 1 [3].

The baseline model just uses word embeddings, an encoding layer, an attention layer, a modeling layer, and finally an output layer. We decided to keep the word embeddings while also concatenating them with character embeddings. We did away with all the other layers, but the new layers we included mimic them in some ways. For instance, the self-matching attention layer encodes the information within the passage. Also, the use of attention is definitely still existent in our model.

The methods for gated attention networks as well as self-matching attention are used in *Gated self-matching networks for reading comprehension and question answering*[7]:

http://www.aclweb.org/anthology/P17-1018

In this paper, a bidirectional recurrent network is used to generate the question-aware passage representation. Then, self-matching is applied on the whole passage to refine the passage representation.

The primary inspiration for this paper is *R-Net: Machine Reading Comprehension with Self-Matching Networks*[6]:

https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf

The model used in this paper involves a gated-attention network to generate question-aware passage representation as well as a self-matching attention mechanism. Pointer networks are also used at the

end to identify the locations of answers in the contexts. We took these key ideas and implemented them as part of our own model.

3 Approach

Our model improves on the baseline model through the use of character-based embeddings and self-attention layers. Specifically, we have patterned our model on R-Net, as described in the following paper [6]:

https://www.microsoft.com/en-us/research/wp-content/uploads/2017/05/r-net.pdf

We begin by computing embeddings for each character in a word. Then, we use a convolutional neural network on the embeddings to gather the combination of the character embeddings for entire words. Then, we run the embeddings through a Highway Network. Finally we apply dropout with a probability of .1 on the network.

After getting these character embebddings, we concatenate them with the embebddings for the overall word using the GloVe vector representations [4].

This gives us new embeddings $c'_1, ..., c'_N$ and $q'_1, ..., q'_M$, an idea adopted from *Bidirectional Attention Flow for Machine Comprehension*[5].

We then use a bidirectional RNN to get new embeddings $c_1, ..., c_N$ and $q_1, ..., q_M$:

$$c_t = BiRNN(c_{t-1}, c'_t)$$
$$q_t = BiRNN(q_{t-1}, q'_t)$$

Next, we use a Gated Attention-Based Recurrent Network to generate context representations that are "question-aware." Let the superscript P denote this awareness. We begin by computing

$$c_t^P = RNN(c_{t-1}^P, [c_t, a_t])$$

where $a_t = att(q, [c_t, c_{t-1}^P])$ are attention-pooling vectors for the whole question for each context word (all $t \in \{1, ..., N\}$). We compute the attention-pooling vectors as follows:

$$d_j^t = c_t^T \tanh(W_1 q_j + W_2 c_t + W_3 c_{t-1}^P)$$
$$s_i^t = Softmax(d^t)$$
$$a_t = q * s^t$$

Then, we add a gate to the input of the RNN:

$$g_t = sigmoid(W_g[c_t, a_t])$$
$$[c_t, a_t]' = g_t \cdot [c_t, a_t]$$

Now, we use $[c_t, a_t]'$ as the input to the RNN instead of $[c_t, a_t]$.

Now that we have new "question-aware" representations of our context $c_1^P, ..., c_N^P$, we will follow the procedure for computing self-matching attention outlined in the paper. We again have an attention pooling vector of the entire context $a_t = att(c^P, c_t^P)$. Then, we use a bidirectional RNN to create the self-attention layer as follows:

$$h_t^P = BiRNN(h_{t-1}^P, [v_t^P, c_t])$$

This attention layer is important in providing passage context for the question-aware passage representation, which allows for syntactic divergence between the question and passage. Moreover, a

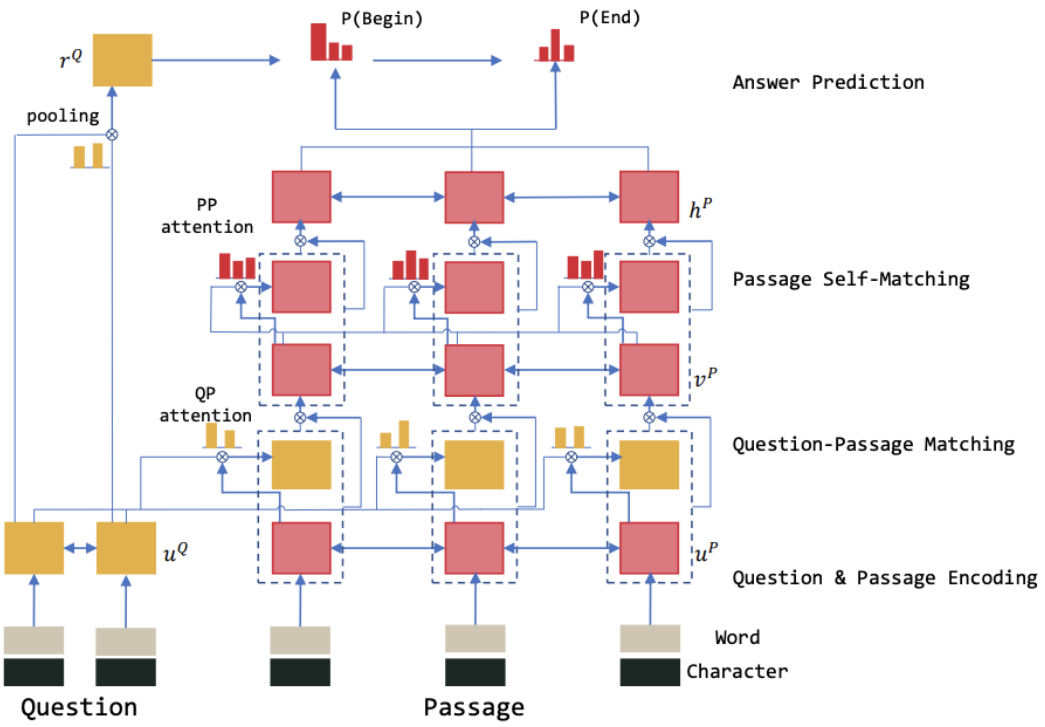

Figure 2: Overall Model Architecture

gate is used here as well to control the input of the RNN.

Finally we use pointer networks to predict where the start and end position of the answer is (p_1 and p_2 respectively). Given our passage representation $c_1^P, ..., c_N^P$,

$$d_j^t = c_t^T \tanh(W_{c_1} c_j^P + W_{c_2} c_{t-1}^a)$$
$$s_i^t = Softmax(d^t)$$
$$p_t = \arg\max_i(s_i^t)$$

Here, c_{t-1}^a represents the previous hidden state of the pointer network, which is computed as follows:

$$a_t = c^P * s^t$$
$$c_t^a = RNN(c_{t-1}^a, a_t)$$

The initial state of this RNN is r_q, which is calculated using the question embeddings:

$$d_j = c^P \tanh(W_q q_j)$$
$$s_j = Softmax(d^t)$$
$$r_q = q * s$$

Figure 2 shows a schematic of the entire model.

	Dev	Test	R-Net	Baseline
EM	57.873	58.445	71.1	55.991
F1	61.504	61.635	79.5	59.291

Table 1: Data

4 Experiments

4.1 Data

The dataset we are using is cited in the Default Final Project Paper. It is the Stanford Question Answering Dataset (SQuAD), a series of context passages, questions regarding the passages, and their answers. Here, we have an example:

Context: Tesla later approached Morgan to ask for more funds to build a more powerful transmitter. **When asked where all the money had gone, Tesla responded by saying that he was affected by the Panic of 1901**, which he (Morgan) had caused. Morgan was shocked by the reminder of his part in the stock market crash and by Tesla's breach of contract by asking for more funds. Tesla wrote another plea to Morgan, but it was also fruitless. Morgan still owed Tesla money on the original agreement, and Tesla had been facing foreclosure even before construction of the tower began.

Question: On what did Tesla blame for the loss of the initial money?

Answer: Panic of 1901

The dataset consists of 129,941 training examples, 6,078 dev examples, and 5,915 test examples. Our task is to answer a series of questions provided by the dataset usnig context passages also provided by the dataset.

4.2 Evaluation Method

Our evaluation method uses F1 and EM scores to compare quantitatively with the scores of other models in the SQuAD challenge. For the purposes of the project, we wanted to obtain scores higher than that of the baseline:

$$EM : 55.991$$

$$F1 : 59.291$$

However, we also wanted to achieve scores close to that of R-Net:

$$EM : 71.1$$

$$F1 : 79.5$$

4.3 Experimental Details

We used a learning rate of 1, a batch size of 16 (for memory purposes) and trained on 20 epochs.

Moreover, the size of the hidden layer used was 50 and we used the 300-dimensional GLoVE vectors, as used by the baseline model.

4.4 Results

We made submissions to the non-PCE leaderboard with the following dev scores:

$$EM : 57.873$$
$$F1 : 61.504$$

We also had the following scores on test:

$$EM : 58.445$$
$$F1 : 61.635$$

These scores showed a marked improvement over the baseline and show that our model generally performs well. However, our score does not reach that of R-Net, which we tried to pattern our model off of.

This tells us that our approach, while still effective, can still be improved significantly. Falling short of R-Net may primarily due to our adjustments to the hyper parameters and layer sizes in order to get the code to run faster.

For example, R-Net used a hidden size of 75 in its model. However, we decided to use 50 to save time and space. Moreover, R-Net ran for 30 epochs, but we had to cut this down to 20 due to time constraints and the limitations of our GPU.

5 Analysis

Consider the following example with an incorrect answer:

Context: Wealth concentration is a theoretical process by which, under certain conditions, newly created wealth concentrates in the possession of already-wealthy individuals or entities. According to this theory, those who already hold wealth have the means to invest in new sources of creating wealth or to otherwise leverage the accumulation of wealth, thus are the beneficiaries of the new wealth.

Question: Who is best able to leverage the accumulation of wealth?

Model's Answer: in the possession of already-wealthy individuals or entities

Correct Answer: those who already hold wealth

This shows us that the model performs poorly on this example in which the answer is in a different part of the context than the area in which keywords of the question exist. Although the model answer contains the correct answer we desire, which is a product of the self-attention layer learning the entire context of the paragraph, it also contains irrelevant text. Namely, it starts with the phrase "in the possession", which is not an answer to the question provided. This may be because the phrasing of the question suggests an answer that is the object of some action. In this case, the object of the sentence would normally follow the word "concentrates". Thus, the model uses this assumption and outputs the answer starting with "in the possession..."

Next, consider the following example with a fully correct answer:

Context: The period from the earliest recorded raids in the 790s until the Norman conquest of England in 1066 is commonly known as the Viking Age of Scandinavian history. Vikings used the Norwegian Sea and Baltic Sea for sea routes to the south.

Question: What did Vikings use as sea routes to the south?

Model's Answer: Norwegian Sea and Baltic Sea

The reason the model was able to get this answer perfectly correct was because the words from the question were explicitly used in the context. Our model builds a question-aware passage representation. This means when the question is easily found within the passage, the accuracy of the

answer is very high.

6 Conclusion

Our project proves to us the effectiveness of self-matching attention on the central task of the SQuAD dataset. While our scores improve on the baseline model, they do not reach the scores of R-Net. This primarily has to do with the hyperparameter tuning we did to make the model perform more efficiently.

In the future, with more time to run the model, we could use the actual model hyperparameters of R-Net, which would undoubtedly be more effective. Moreover, we can experiment with the effectiveness of certain layers or parts of layers by simply removing them and seeing how well the model still performs.

Acknowledgments

We would like to thank Prof. Manning and all the staff of CS 224N for all their help teaching us the material necessary for working on this project as well as providing us with the idea for a wonderful default project.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding (Google Language AI). arXiv:1810.04805. 2018.

[2] Ming Zhou, Nan Duan, Furu Wei, Shujie Liu, and Dongdong Zhang. The Next 10 Years Look Golden for Natural Language Processing Research (Microsoft Asia Research). 2018.

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

[4] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pages 1532–1543, 2014.

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

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

[7] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In Association for Computational Linguistics (ACL), 2017.