Self-Attention in Question Answering

Stanford CS224N Default Project

William Wang
Stanford University
wjwang@stanford.edu

Abstract

The SQuAD challenge is a task in which models are asked to perform a question answering task over the Stanford Question Answering Dataset. To do so, our model will improve on the baseline BiDAF model's F1 and EM scores by using mechanisms used in R-NET, which performs better on the SQuAD dataset than the baseline BiDAF model. In this paper, we will discuss two mechanisms used in R-NET that we applied to baseline model: the addition of character embeddings and the addition of a self-attention layer.

1 Introduction

Over the past several years, the use of deep neural networks has led to the development of many new models that have allowed us to make large improvements in different tasks over a variety of fields, including natural language processing. In particular, the task of contextual question answering has seen large improvements over the past decade. Specifically, the development of models that can successfully take large bodies of text and extract answers to questions based on these texts could be valuable in a variety of applications, and can even be extended to other tasks such as relationship extraction[1].

With the amount of new models and new techniques being applied every year, there needs to be some method or benchmarks to compare them against. The Stanford Question Answering Dataset, or SQuAD, is a dataset that has increased in popularity over the past few years as it allows for making such benchmarks. It is a dataset where models are given a question and a respective context paragraph and are expected to generate an answer by extracting some span of the context paragraph.

Moreover, there are two different versions of the dataset: SQuAD1.1 and SQuAD2.0. SQuAD1.1 consists of over 100,000 question-answer pairs where models are always expected to extract an answer, even though it might be the case that the answer was provided in the context paragraph. To account for this SQuAD2.0 adds 50,000 unanswerable questions so that models must be able to also determine when the answer was not provided in the context paragraph, in which case it should abstain from answering. In this sense, SQuAD2.0 requires models to be more robust than those built based on SQuAD 1.1.

State of the art models have been built that exceed human levels of performance on both SQuAD datasets. A large part of this can be attributed to the development of transformer-base models such as BERT[2]. Today, most of the current work being done with the dataset is with the SQuAD2.0 dataset. For our models, we will also be working with the SQuAD 2.0 dataset.

2 Related Work

The baseline for this model is based on BiDAF [3]. The original paper's model, which is shown in Figure 1, consists of an embedding layer, an encoding layer, an attention layer, a modeling layer, and an output layer. The baseline implementation differs from the original paper's as the embedding layer consists only of word-level embeddings and not character-level embeddings.

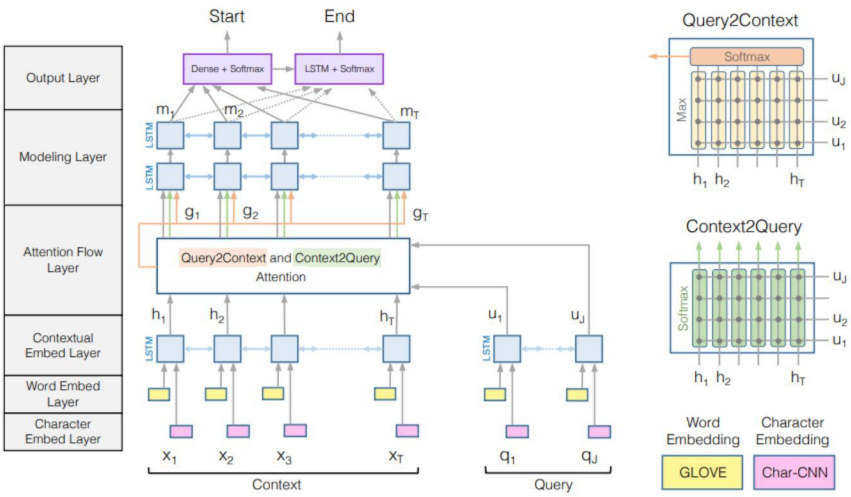

Figure 1: BIDAF Model

Another question-answering model is R-Net [4], whose model is shown below in Figure 2. R-Net also consists of an embedding layer that uses both word-level embeddings and character-level embeddings. It then has a gated attention layer which generates an attention vector that attends between the question and context. Lastly, it has a self-attention layer and an output layer that predicts the stat and end of the answer using pointer networks.

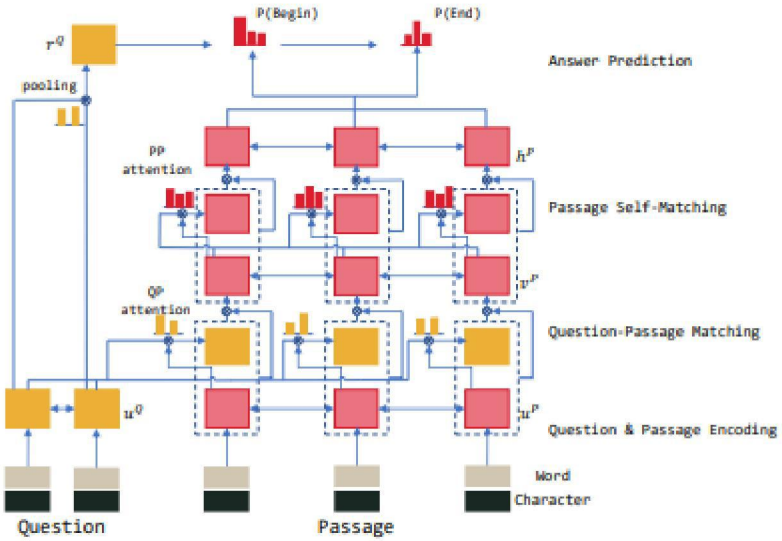

Figure 2: R-Net Model

The most important aspect of the R-Net model is the self-attention layer. This is because the vectors generated from the gated attention layer are limited in their amount of knowledge in the surrounding context. However, as far away context is often necessary to extract the answer, self-attention allows the representation to capture context from the whole paragraph rater than just the limited surrounding context. This idea of self-attention is a key idea in Transformer models such as QANet [5] that have become increasingly popular in question-answering and other natural language processing tasks as well.

3 Approach

Our goal for our model was to take some mechanisms from previous models and apply them to our baseline. The first addition we wanted to make was to use character embeddings along with the word embeddings, similar to both the original BiDAF and R-Net. The second addition was to add a self-attention layer, similar to the one described in R-Net. For computational reasons, we also changed the RNN in the encoding layer from a LSTM to a GRU. The general structure of our model is shown below in Figure 3.

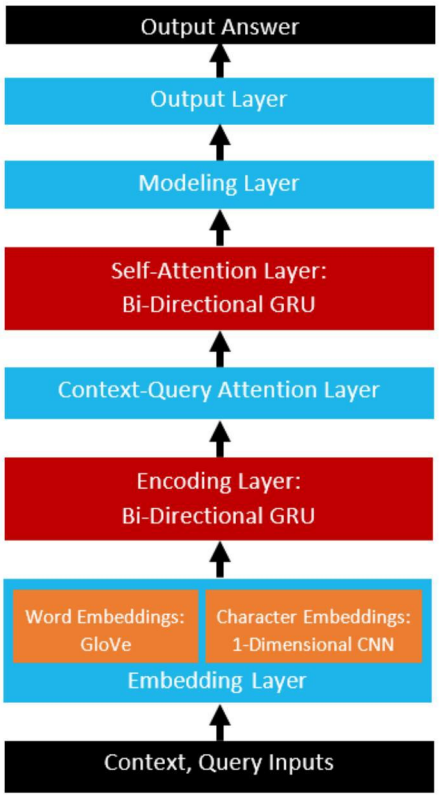

Figure 3: Overall Model Architecture

3.1 Embedding Layer

To generate character embeddings, we used the approach described in the BiDAF paper. Namely, for each word, we use a 1-dimensional convolutional neural network (CNN). We then used max pooling to ensure that the tensor size was fixed for each word in our vocabulary. Lastly, we concatenate these character embeddings with our word-level GloVe Vector embeddings [6], and then pass these new embeddings into a highway encoder.

3.2 Self-Attention Layer

For our self-attention layer, we use one similar to the one described in R-Net. Given our attention-based outputs g_t from our context-query attention layer, we will perform the following equations to apply additive attention:

$$e_i^t = v^T tanh(W_1 g_t + W_2 g_t)$$
$$a_i^t = softmax(e_i^t)$$
$$c_t = \sum_{i=1}^{n} a_i^t g_i$$

Here, note that after multiplying by our weight vectors, we unsqueeze them along different dimensions. Furthermore, similar to R-Net, we concatenate c_t to our previous attention. After feeding this into our GRU and applying dropout, we obtain a "self-aware" representation, which we then feed into our modeling layer.

4 Experiments

4.1 Data

We use the SQuAD 2.0 dataset as mentioned earlier. The dataset consists of training examples that are comprised of context paragraphs, questions, and answers that are a continuous span of the context paragraph. A training example is given below:

Context paragraph: On 24 March 1879, Tesla was returned to Gospic under police guard for not having a residence permit. On 17 April 1879, Milutin Tesla died at the age of 60 after contracting an unspecified illness (although some sources say that he died of a stroke). During that year, Tesla taught a large class of students in his old school, Higher Real Gymnasium, in Gospic.

Question: Why was Tesla returned to Gospic?

Answer: not having a residence permit

The dataset has already been split into train, dev, and test sets. The train set has 129941 examples, the dev set has 6078 examples and the test set has 5915 examples. Our task for this dataset is to answer questions based on some given context paragraph or indicate that the context paragraph does not contain the answer.

4.2 Evaluation method

We evaluate our results using 2 metrics. The first is the F1 metric, which is defined to be the harmonic mean of precision and recall. Specifically, precision is the true positive count divided by the number of times the system return a non-null answer while recall is the true positive count divided by the number of instances that have an answer. The second metric is EM, which represents exact match. The EM metric evaluates if the predicted answer exactly matches the ground truth.

4.3 Experimental details

When initially training our baseline, we initially used a batch size of 64 and a hidden layer size of 100 and trained for 30 epochs. However, after adding on a self-attention layer, we ran into memory issues when using the same hyper parameters. Because of this, we reduced the batch size of 16 and the hidden size layer size to 25.

For consistency with our baseline model which was trained using a dropout rate of 0.2 and learning rate of 0.5, we kept these hyper parameters the same when training our model with self-attention. These hyper parameters were similar to R-Net, which also used a dropout rate of 0.2 and an initial learning rate of 1.

4.4 Results

Model	Exact Match	F1
Baseline	57.789	61.122
Final Model w/ Reduced Hyper Parameters (test)	60.237	63.942
R-NET Single Model	71.1	79.5

These scores show that adding character embeddings and a self-attention layer makes improvements over our baseline. This was to expected as character embeddings gives our model more representational power and allows us to better handle words not defined in our word vocabulary while a self-attention layer allow our model to better capture long-distance dependencies across context passages, which we hope will make further improvements with these evaluation metrics.

However, these scores are still significantly lower than some other similar models such as R-Net. Most likely, improvement can be made through fine tuning of hyper parameters as we had to reduce both batch size and hidden layer size due to memory constraints of our GPU. For reference, R-Net had more representational power as it used a hidden size of 75 instead of 25.

5 Analysis

Even with an additional self-attention layer, there are still several limitations of the model. Consider the following examples where the model returns an incorrect prediction.

Context paragraph: Instability troubled the early years of Kublai Khan's reign. Ogedei's grandson Kaidu refused to submit to Kublai and threatened the western frontier of Kublai's domain. The hostile but weakened Song dynasty remained an obstacle in the south. Kublai secured the northeast border in 1259 by installing the hostage prince Wonjong as the ruler of Korea, making it a Mongol tributary state. Kublai was also threatened by domestic unrest. Li Tan, the son-in-law of a powerful official, instigated a revolt against Mongol rule in 1262. After successfully suppressing the revolt, Kublai curbed the influence of the Han Chinese advisers in his court. He feared that his dependence on Chinese officials left him vulnerable to future revolts and defections to the Song.

Question: Who was Kaidu's grandfather?

Answer: Ogedei

Model Prediction: Li Tan

This example demonstrates that the model does not perform as well when the key words in the question query are not also provided in the context. Specifically for this example, while the question asks about Kaidu's grandfather, the word "grandfather" is not given in the context paragraph. Instead, since the context paragraph describes "Ogedei's grandson Kaidu", the model must successfully extract the relationship between "grandfather" and "grandson" in order to return the correct answer.

However, the model instead incorrectly "Li Tan" who was "the son-in-law (of a powerful official)". This suggests that perhaps the model better understands the relationship between "father" and "son" as this relationship would be more likely to show up in training examples compared to the relationship between "grandfather" and "grandson". For this reason, the model focuses instead of this relation and outputs the incorrect answer "Li Tan" instead of "Ogedei",

Now consider another example where the model returns an incorrect prediction:

Context paragraph: The war was fought primarily along the frontiers between New France and the British colonies, from Virginia in the South to Nova Scotia in the North. It began with a dispute over control of the confluence of the Allegheny and Monongahela rivers, called the Forks of the Ohio, and the site of the French Fort Duquesne and present-day Pittsburgh, Pennsylvania. The dispute erupted into violence in the Battle of Jumonville Glen in May 1754, during which Virginia militiamen under the command of 22-year-old George Washington ambushed a French patrol.

Question: How did peace start?

Answer: N/A

Model Prediction: a dispute over control of the confluence of the Allegheny and Monongahela rivers

This example again shows us a similar pattern with our model as the previous example. In this instance, the context paragraph describes a certain war but the question asks for how peace, the opposite of war, started. However the word "peace" does not appear at all in the context paragraph. Instead the model likely captures the relationship between the word "start" and "began", which is why it predicts the answer "a dispute over control of the confluence of the Allegheny and Monongahela rivers", which would be the answer to the question "How did war start?" instead of peace. This example shows that the model doesn't handle antonyms very well, which could potentially be used adversarially against some of these models.

6 Conclusion

Our project shows us that the addition of character embeddings and self-matching attention to BiDAF models can help improve on the task of question-answering. However, our model does not reach the same scores as other models such as R-Net. This is likely due to our limitations in memory and computation time, which led to reductions in some of the hyperparameters.

In the future, given larger memory, we would like to test our model with self-attention using larger batch size and hidden layer sizes. Additionally, with more time, we would've also liked to test out the effect of changes in dropout or learning rate.

Overall, this project has provided us with valuable knowledge and first-hand experience in training deep neural networks and understanding recent deep learning models. We would like to thank all of the teaching staff of CS 224N for teaching us about natural language processing and providing us the necessary tools and support for this project.

References

[1] Omar Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. 2019.

[3] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bi-directional attention flow for machine comprehension. 2016.

[4] Natural Language Computing Group Microsoft Research Asia. R-net: Machine reading comprehension with self-matching networks.

[5] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. 2018.

[6] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. 2014.