R-NET Prime

Nik Caryotakis,[1] Parker Killion,[1] J.P. Reilly[1]

[1]Department of Computer Science, Stanford University

Stanford
Computer Science

Abstract

Building upon the Bidirectional Attention Flow model (BiDAF) and inspired by R-NET we implemented R-NET Prime. Featuring character level embeddings and a pointer network for an output layer, R-NET Prime improves the performance of the baseline model by using self attention in addition to the bidirectional attention layer in BiDAF. Our version of self attention is more easily parallelizable as opposed to R-NET's, allowing for faster training. This faster training allowed us to perform an ablation study to isolate the performance contribution of each R-NET Prime component, as well as thorough hyper parameter tuning.

We evaluate R-NET Prime on the SQuAD 2.0 dataset, and achieve an F1 score of 63.67 and EM score of 60.37 , placing our team, Palo Alto High School, at 42nd on the CS 224N leaderboard at the time of writing.

Stanford
University

Introduction

The SQuAD dataset was developed by Stanford Researchers and houses a leader board for models to be compared. In this project, we investigate R-NET a model that topped the leaderboard when it was released. The R-NET utilizes self-attention and we wanted to investigate how effective self-attention is in their model and see if we could improve upon it in our own. Our implementation, R-NET prime, utilizes character embeddings, self-attention, and a pointer-network output layer, in addition to several features from the given BiDAF starter code. We describe the SQuAD task below:

Given: A question, and a context paragraph.
Output: The span of text that answers the question.

Example:

Question: Why was Tesla returned to Gospic?
Context: 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.
Answer: not having a residence permit

Approach

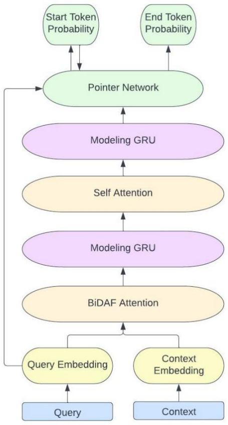
We started by running the baseline model given to us in the SQuAD starter code which is a Bidirectional Attention Flow model. We extend upon this model in three ways: character-based embeddings, self-attention, and a pointer network output layer.

Approach (cont.)

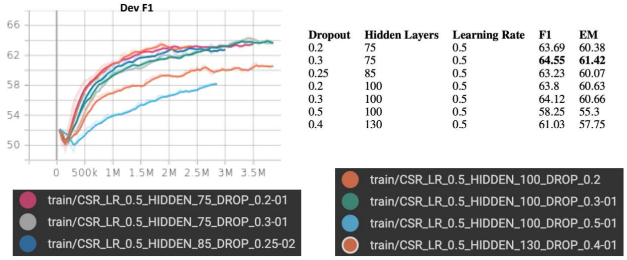
Character-level embeddings are found by taking final forward and backward hidden states of a bi-directional recurrent neural network (RNN), which accepts pre-trained letter embeddings one at a time. These two states are concatenated together to form a character-level word embedding, that is then concatenated with the 300 dimensional GloVE word vectors given.

We add a self attention layer and an additional modeling layer to BiDAF. For self attention, different from R-NET's paper, we do not use the output of one RNN cell as part of a gated input into the next. The output from this layer is the concatenation of the original context representation, the attention weights, and the two multiplied together. This is then fed into a modeling layer, just as the BiDAF attention does before it.

Third, we replace the BiDAF output layer with a Pointer Network (Ptr-Net) [6], continuing to follow the R-NET paper. This network uses attention as a way to select input tokens that work best as start/end indices for our question answering system.



Results



Dropout	Hidden Layers	Learning Rate	F1	EM
0.2	75	0.5	63.69	60.38
0.3	75	0.5	**64.55**	**61.42**
0.25	85	0.5	63.23	60.07
0.2	100	0.5	63.8	60.63
0.3	100	0.5	64.12	60.66
0.5	100	0.5	58.25	55.3
0.4	130	0.5	61.03	57.75

Analysis

We performed an ablation study to analyze the unique effects and impacts separate pieces of our R-NET Prime model, and then to finally determine the effect our faster, modified attention mechanism had in comparison to other standard R-NET layers. We created a naming system for the models where (B) stands for baseline, (S) stands for self attention, (C) stands for character embeddings, and (R) stands for R-NET output pointer-network. The table below shows our results:

Model Type	Dropout	Hidden Layers	Learning Rate	F1	EM
BBB	0.2	100	0.5	60.94	57.96
CBB	0.2	100	0.5	62.74	59.57
BSB	0.2	100	0.5	62.57	59.2
BBR	0.2	100	0.5	60.92	57.3
CSR	0.2	100	0.5	63.8	60.63
CBB	0.5	100	0.5	56.76	53.71
BSB	0.5	100	0.5	62.57	59.2
BBR	0.5	100	0.5	56.34	53.96
CSR	0.5	100	0.5	58.25	55.3
CBB	0.2	75	0.5	62.86	59.6
BSB	0.2	75	0.5	61.2	57.84
BBR	0.2	75	0.5	60.31	57.13
CSR	0.2	75	0.5	**63.69**	**60.38**

Typically character level embeddings outperforms both self-attention and the pointer network when ran in isolation. The pointer network and character embeddings suffer greatly from the higher drop out. The pointer network appears to only have marginal increase in our model. Overall, R-NET Prime framework gains the most from the character level embeddings as it encodes more information for the rest of the model to gather from.

References

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

Seo Minjoon, Aniruddha Kembhavi, Ali Farhadi, and Hananneh Hajischrizi. Bi-directional atten- tion flow for machine comprehension. In *International Conference on Learning Representations*, 2017.

Natural Language Computing Group. R-net: Machine reading comprehension with self-matching networks. May 2017.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks, 2017.