All SQUAD Needs is Self-Attention: Char-QANet
Luís Alcaraz, Troy Lawrence
Department of Computer Science, Department of Electrical Engineering
Yian Zhang

Introduction

Techniques such as Long-Short Term Memory (LSTM) networks were able to improve on previous methods that used separate learning, separate translation, and separate argument generation. However, the problem of mapping the meaning of long sequences of data to real-time and context-dependent information is still unsolved. Self-attention and a single, shared, architecture, like AlphaNet, was able to solve this problem. However, AlphaNet architecture is unique and requires specific tools. Therefore, we wanted to try out building a transfer-like architecture. QANet gives a possible solution to this problem. Performing on top of a classical sequential full supervised model, QANet outperformed all competing models on the SQuAD dataset in terms of exact match and the learning efficiency. However, there is a learning efficiency gap between sequential and convolutional layers.

Although, the basic attention mechanism is powerful and has been successfully used in various tasks, it has some limitations. Therefore, we propose a novel self-attention mechanism that addresses some of these limitations. We experiment with QANet, allowing us to get a better understanding of this architecture and its limitations.

Key Findings

- QANet and the self-attention mechanism are very powerful. Due to the multi-head attention, the model was able to create local and global dependencies that surpassed LSTM’s and AlphaNet’s approach. This was also done with one third less epochs.

- However, we also learned that QANet has its limitations, which it shares with transformers. Due to these blocks that consist of convolutional layers and self-attention layers, the model becomes memory expensive, which comes to limit which hyperparameters one can choose to obtain the best results. Such challenges forced us to limit our batch size from the recommended 64 to 16 right from the start.

- Nonetheless, we are still hopeful in the use of transformers and models like QANet which utilize transformer building blocks advanced models towards making transformers more memory efficient.

Methods

Our implementation started with character embeddings, which allow OHs to create the proposed character-attention. We also showed attention to our baseline model which has a strong performance on the baseline. All data from SQuAD 2.0 with a split into training, dev, and test set. All input embeddings were used as input embedding layer.

We build a character-attention which is a part of the baseline model. By using medical encoder blocks, we allow the medical team complete dependencies. By using embedded character order and character embedding through convolutional neural networks, we achieve best features of both.

We evaluate character-attention and character-attention on the baseline model which allows the model to learn global dependencies between words.

We evaluated character-attention and character-attention on the baseline model, which allows the model to learn global dependencies between words.

We also experimented in these hyperparameters and identified potential drawbacks from QANet.

Analysis

One of the first things we identified was the importance of hyperparameters. This was visually apparent when choosing the proper optimizer. The baseline utilizes Adabound optimizer in comparison to QANet (baseline). This, we believe, is due to Adam’s bias correction towards moments, leading to a faster convergence.

Furthermore, the scheduler was also extremely important in QANet requires an increase exponential decrease. In addition, we found that the learning method was set at 100x. We ran experiments for learning an increasing log-weighted loss towards exponential increase within 5 epochs. Selecting it to QANet’s suggested schedule increased results.

References


Acknowledgments

(C)2019 Training staff for an unprecedented set of learning through new experiences and inspirational instruction.