Solving SQuAD 2.0 with QANet

Jane Boettcher [1] Matthew Kaplan [1] Jacklyn Luu [2]

[1]Department of Computer Science, Stanford [2]Department of Biomedical Informatics, Stanford

Problem

Machine reading comprehension is a central task in natural language understanding. To this end, SQuAD offers a large number of questions and answers created by humans through crowd-sourcing [1]. Our task is to create a question answering system that works well on SQuAD 2.0, which extends the original data set with unanswerable questions. As input, a model will be given a paragraph, and a question about that paragraph. The goal is to predict the correct answer within the paragraph if it exists.

Methods

BiDAF

Our baseline model BiDAF combines RNN encoders with attention to predict the start and end points of an answer within a passage. We added a convolutional layer applied to character embeddings to increase the score of the base model, which previously used just word embeddings.

QANet

QANet is a transformer model based on stacks of encoder blocks, where each block includes a convolutional layer, self-attention layer, and a feed-forward net layer . The model uses the same context-query attention, and word and character embeddings as BiDAF. For the feed-forward net in the Encoder Blocks, we use a simple 2-layer MLP (Linear → ReLU → Linear). [2]

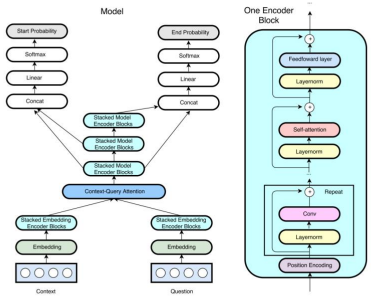

Figure 1. QANet architecture

Experiments

We trained the following models on SQuAD 2.0:

1. Baseline BiDAF model
2. BiDAF model with character embeddings
3. QANet with 5 blocks in the model encoder block stack and no layer dropout
4. QANet v2 with 7 blocks in the model encoder block stack, an added ReLU nonlinearity in each encoder block, and layer dropout after each layer within the encoder block as described by [2]

For all models, we used the Adadelta Optimizer with a learning rate of 0.5 and no weight decay. For both QANet models, the embedding encoder block stack consists of just 1 encoder block.

	Dev Set		Test Set	
Model	F1	EM	F1	EM
BiDAF (baseline)	61.38	57.94		
BiDAF + char embeddings	64.83	61.42		
QANet (5 Blocks)	65.19	61.94	63.97	60.54
QANet v2 (Full Model)	**68.15**	**65.13**	**66.05**	**62.87**

Table 1: Performance of various models on SQuAD 2.0 dev and test sets

Analysis

First, we examined how our models performed on questions with different types of question words as seen in Figure 2. We found that "why" questions are hardest for our models to answer and "when" questions are easiest.

Figure 2. Performance of our models broken down by question words

Furthermore, question words like "what" and "which" can easily overlap with "where" if phrased like "what country". We used part-of-speech tagging and named-entity-recognition, to re-categorize questions into question categories. We similarly found that description questions are hardest to answer, and predict this might be because description questions require more contextual understanding and don't have a typical answer format.

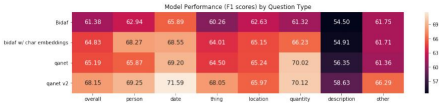

Figure 3. Performance of our models broken down by question types

Analysis cont.

Finally, we categorized questions by their answers in cases where answers were entities. We find that most named entity categories performed better than non-named entities (categorized as "OTHER"). This may be because these named entities are more concrete or distinctive and can also take very distinctive formats.

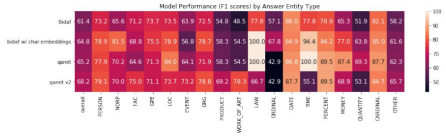

Figure 4. Performance of our models broken down by answer entity types

Discussion

Overall, we found QANet to outperform BiDAF, even without layer-dropout and only 5 encoder blocks in the model encoder block stack. Our best model came from following the original authors' QANet architecture specifications (namely, 7 encoder blocks in the model encoder block stack and layer dropout).

An interesting observation is that our QANet model did not fully match the performance that the authors achieved in the original paper. This may be attributed to the use of SQuAD 2.0 instead of the original SQuAD dataset; the original authors' use of data augmentation to generalize the training set (which we did not use in this project); or our use of a simple 2-layer MLP for the feed-forward net in the encoder blocks instead of a more elaborate architecture.

Further, we see that our model seems to do better on questions that have a clear expected answer format and worse on questions that prompt for descriptions.

Conclusions

We were able to build our own version of QANet, which performed far better than the baseline BIDAF on answering questions. Our model works especially well with questions that ask about dates and quantities and worse with questions that ask for some sort of description or explanation.

References

Poster template provided by [3]

[1] Pranav Rajpurkar, Robin Jia, and Percy Liang.
Know what you don't know: Unanswerable questions for SQuAD.
In Association for Computational Linguistics (ACL), 2018.

[2] Minh-Thang Luong Rui Zhao Kai Chen Mohammad Norouzi Quoc V. Le Adams Wei Yu, David Dohan.
QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension.
2018.

[3] RylanSchaeffer.
Stanford-LaTeX-Poster-Template. https://github.com/RylanSchaeffer/Stanford-LaTeX-Poster-Template.