Exploring CNNs in Question Answering

Stanford CS224N Default Project

Joseph Pagadora, Steve Gan
Stanford University
jcp737@stanford.edu zgan@stanford.edu

1 Research paper summary

Title	QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension
Authors	Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, Quoc V. Le
Venue	ICLR 2018
Year	2018
URL	https://arxiv.org/pdf/1804.09541.pdf

Background. This paper introduces a new architecture (QANet) to question-answering completely without the use of RNNs **?**. Many of the prior works toward this problems involve the use of RNNs and their ability to encode sequential data, but as a consequence, this results in longer training and inference time. At the time when RNNs with attention were the best way to learn from language data, this paper proposed to use convolutional layers with self-attention in question-answering models to achieve fast training and inference. The higher speed makes a training on a larger data-set feasible, and so a data-augmentation method via backtranslation is explored.

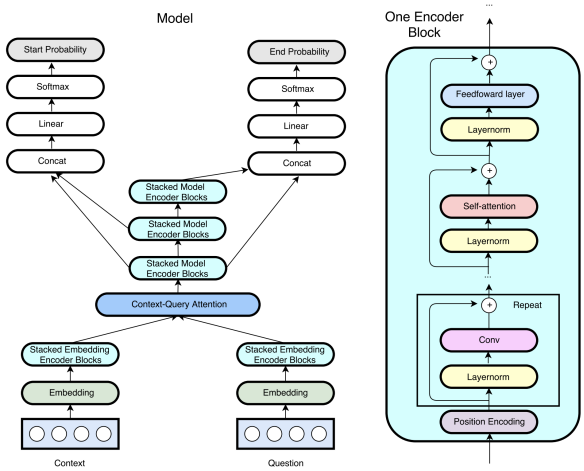

Figure 1: Left: Architecture of QANet with convolution and self-attention. Right: Encoder block that is used throughout the model.

Summary of Contributions. This paper proposes a new and faster way to approach the task of reading comprehension and question-answering without the use of a recurrent model. This approach

is motivated by slow training and inference of RNNs. The paper outlines a new method that uses encoders consisting only of convolutional layers for local analysis of the input, along with self-attention for global analysis. This resulted in an increase in training speed about 3 to 13 times and in inference speed about 4 to 9 times compared to the standards set by previous high-accuracy methods that used RNNs.

Because of this drastic increase in efficiency, the model is able to be trained on a much larger dataset. Using the SQuAD dataset, the authors devised a data-augmentation approach: they translated the data to another language, and then translated it back to English, obtaining a paraphrase of the original data. Obviously, they hoped for an imperfect translator, that is, one that does not provide an invertible mapping. In other words, the synthesized datapoint has different phrasing than that of the original. This technique allows the model acquire more data thus improving performance on the question-answering task. The authors were able to achieve a 84.6 F1 score on the SQuAD test set after training on the augmented SQuAD training set.

Figure 1 shows a summarized diagram of the architecture of QANet. The raw input is given as (C, Q) = (context, question) pairs, where the context $C = \{c_1, ..., c_n\}$ consists of, say, n words, and the model is to output a consecutive span $S = \{c_s, ..., c_e\} \subseteq C$ containing the answer to the question Q. The QANet model itself outputs softmax probabilities of each word in the context being the start and end positions s, e, respectively. The first step is to embed the context and question using standard embedding procedures such as GloVe. Then, these embeddings are put through multiple embedding encoder blocks. Figure 1 shows that each encoder block consists of three residual blocks, each containing a convolutional layer, a self-attention layer, and a feed-forward layer, in order. Then, the encoded context and question are put through a standard context-query attention layer. After this, the attention output is put through three model encoders M_0, M_1, and M_2. With learnable parameters W_0 and W_1, each position i in the context is given a start and end probability

$$p_1(i) = \text{softmax}(W_1[M_0; M_1]), \quad p_2(i) = \text{softmax}(W_2[M_0; M_2]),$$

where $[\cdot\ ;\cdot]$ denotes concatenation. Using dynamic programming, the output is a start- and end-position (s, e) that maximizes $p_1(s)p_2(e)$.

Limitations and Discussion. One major limitation of this paper lies in the data augmentation step. In particular, the authors use a beam decoder in both the forward (to foreign language) and backward (back to English) translation to obtain a set of possible paraphrases for each sentence. The final representation of the original sentence is chosen at random from this set of sentences. After a paraphrase of a document is determined, a candidate phrase in the document is chosen to be the answer by computing the character-level-2-gram scores relative to the original answer. One issue is that the quality of the augmented data is not as high as the original data. Another issue is that the quality of the augmented data set can be enhanced if the selection process for each paraphrased sentence incorporates information about the original answer. Other approaches to paraphrasing such as the type swap method **?** can be explored.

Why This Paper? We chose this paper primarily because its focus on CNN. In this NLP course, we have mostly learned about RNN models, and we are interested in models that can readily be parallelized and yield faster training and inference speed. We have gained insights into an alternative way to construct a language system as there is no time step involved.

Wider Research Context. This work contributes to the field of NLP in two ways. First, it proposes a new system architecture for question-answering models. RNNs with attention has been found effective at performing the question-answering task **?**. RNN is a sequential modeling architecture that can capture the contextual information of the input while attention makes the model to focus on certain part of the input at each time step **?**. Vaswani et al. proposed a transformer model architecture for NLP tasks that entirely discards the sequential layer and is entirely based on the attention mechanism (self-attention) **?**. The current paper uses the bidirectional attention between query and context found in Seo et al.'s work and feed the output of this layer to contextual layer using CNN and self-attention.

Second, the use of backtranslation for data-augmentation for question-answering models is also new. Backtranslation had been found effective in data-augmentation for neural machine translation when the target data is monolingual and the target language has scarce data **?**. The SQuAD dataset can also be considered as scarce in the sense that all models that want to compete on their performance on this dataset have to use the same training set.

2 Project description

Goal. We will investigate whether changing the backtranslation pipeline would improve the performance of the QANet model. We will first implement the character-level embedding layer (also a CNN layer) in BIDAF. Then we will build a QANet using the scaffolding given in the default model. We then implement the type swap method as the data-augmentation component. If we have time, we will also implement the original backtranslation pipeline and the variant we suggested in the Limitations and Discussion section. Then we can compare the performance of the BIDAF without the character-level embedding layer, BIDAF, QANET with type and swap, and QANET with backtranslation.
In essence, we plan to implement QANet in PyTorch, mostly following the model described in detail in the corresponding paper, but with the discussion about the issues/limitations in mind, explore different ways we can modify the approach to data augmentation to better improve the performance of QANet. Furthermore, we plan to explore more ways to combine different ideas to improve performance that are independent of the transformer/CNN structure. A few techniques that interest us are (1) span representation, and (2) changing the details of the self-attention mechanism. These were described briefly in the IID SQuAD track handout. In summary, observe that QANet learns the start and end positions of the answer in the context independently, and it calculates the probability of a span as the product of the probabilities of the start and end positions. Thus, we can instead modify QANet to learn directly the probabilities of the spans (Dynamic Chunk Reader). There are several ways of computing attention: in class, we learned about additive, multiplicative, and dot product. Finally, we can simply play around with basic ways to improve performance such as regularization, weight sharing, changing parameters, ensemble learning, etc.

Task. We aim to build an efficient CNN-based question-answering model and implement a data augmentation method. The task has been described in detail in section 1.

Data. We will use the default SQuAD dataset and the specified preprocessing methods. Note, however, that we will be using a downloaded neural machine translator to perform data augmentation. Thus, in practice we will be using much more data than just the SQuAD dataset, but this extra data does indeed come from the original dataset.

Methods. We will implement the character-level embedding layer and the QANet. The neural machine translation part of the backtranslation pipeline in QANet will be downloaded. As mentioned above, we primarily plan to use convolutional layers and self-attention as described in the paper in section 1. This method is inspired by the idea of transformers, so we will keep that in mind when looking for ways to improve performance of QANet.

Baselines. For the data augmentation, which involves translation, we will download a NMT. The type and swap pipeline will be built from scratch, and we will implement a baseline QANet along with the data augmentation technique which is outlined in detail in the paper. Clearly, this should provide a baseline for our experimental work.

Evaluation A common way to evaluate the performance of a question-answering system is to use the exact-match and F1 scores. The exact-match score only counts outputs that are exact matches to the labeled span, while the F1 score is less stringent by calculating the harmonic mean of the precision and recall of the outputs against the labeled span.