Ch-QANet, a non-recurrent structure model of Chinese machine reading comprehension

YAN Xu, FANG Yueran, ZHANG Tianyang
The Chinese University of Hong Kong, Shenzhen
218012048@link.cuhk.edu.cn

Abstract

Currently, the applications of English reading comprehension based on deep learning got obvious success, but there were not application of Chinese machine reading comprehension can be obtained because of the lack of dataset. Meanwhile, most reading comprehension tasks use recurrent neutral networks (RNNs), so they are often slow for training and inference, such as the most successful model Bi-Directional Attention Flow network (BIDAF). In this paper, we attempt to improve a new architectures called QANet on DuReader, a new Chinese dataset contains Baidu zhidao and Baidu search. QANet's encoder consists only of convolutional and self-attention structure, which can accelerate the training speed by several times. By implementing QANet in Chinese version (we call Ch-QANet in the following parts), we experimented with the best algorithm of BIDAF on the DuReader. Through experiments, we found that Ch-QANet not only surpassed BiDAF in speed, but also performed well in effect.

1 Introduction

The tasks of machine reading comprehension (MRC) and Q&A have made remarkable achievements in the English natural language processing and computer vision community over the past few years. One of the key factors is the attention mechanism[10], which enables the system to focus on the target areas within contexts (for MRC) or images (for Q&A). There are other representatives of using the improved attention mechanism[10], such as Bi-Directional Attention Flow (BIDAF)[7] and QANet[13], which all achieved excellent results in the English datasets. However, there is rarely a complete Chinese-oriented Q&A system dataset in these years, while the existing English datasets, NewsQA[9], SQuAD[6] and TriviaQA[6], have many significant problems: artificial synthetic data, simple task and limited application field.

In this paper, the dataset we use is the large-scale Chinese MRC dataset released by Baidu in 2017-DuReader[3]. This dataset has following features compared with the previous datasets: questions and answers all comes from reality rather than artificial mark, the problem form is more varied and the language is more complicated. Baidu use two baseline models, Match-LSTM[11] and BiDAF[7], on DuReader[3] at the beginning of the release of this dataset, but there are no any experimental results of using QANet. Therefore, our research will focus on training and improving Ch-QANet on this dataset and compare with the Baidu's baseline. In addition, we have add additional multi-head attention mechanism in the model's decoder, making the model perform better in the experiment.

The key contributions of our work are as follows:

- We apply a new Chinese MRC model called Ch-QANet on DuReader.
- Our result can use as a new baseline on this dataset to a certain extent.

1.1 Related Work

New Chinese MRC dataset

Rrecent research shows that existing MRC models are able to achieve high performance on these datasets, but the understanding and inference of these models is limited. Therefore, establishing a real-world MRC data set is urgent and important. Fortunately, Baidu released a large-scale Chinese MRC dataset in 2017-DuReader[3], which has the following characteristics: (1) All questions and texts are derived from real data (Baidu Search and Baidu Zhidao), and the answer is answered by natural human. (2) The data set contains a large number of point of view problem that were rarely covered by othe dataset before. (3) Each question corresponds to multiple answers. The dataset contains 200k questions, 1000k original text and 420k answers. DuReader[3] is the largest Chinese MRC dataset at present, which annotated answers based on questions and documents. In terms of data complexity, the

Under review in the final project for CIE6032 (MDS6232). Do not distribute.

average word length of the problem is 4.8, the average word length of the answer is 69.6, and the average word length of the document is 396.0, which is 5 times that of MS-MARCO[5]. Due to the large scale and complex types of problems, the analysis work based on the DuReader dataset[3] is much more difficult than the previous datasets.

Up-to-date MRC model

In the past few years, the most successful model has usually adopted two key technologies: (1) a recurrent structure to deal with sequential input (2) attention factors to deal with long-term interaction. The Bidirectional Attention Flow (BiDAF)[7] successfully combines these two elements, and the model has achieved remarkable results on SQuAD dataset[6]. One disadvantage of these models is that their recurrent structure makes them generally inefficient in training and reasoning, especially for long texts. Expensive training will not only prolong the experimental period and restrict researchers from fast iteration, but also make it difficult for the model to be used in a larger data set. In order to make machine understanding more efficient, Google proposed to remove the recurrent structure of these models and call it QANet[13], which only uses convolutional and self-attention[10] as the constituent modules of the encoder to encode the problem and context respectively. In SQUAD dataset[6], the training speed of QANet[13] is increased by 3 to 13 times, and the reasoning speed is increased by 4 to 9 times, at the same time, it achieves the accuracy comparable to that of the recurrent model.

2 The Proposed Algorithm

2.1 Problem formulation

The Chinese reading comprehension tasks considered in this paper, is defined as follows. Given a context paragraph with n words $C = \{c_1, c_2, ..., c_n\}$ and the query sentence with m words $Q = \{q_1, q_2, ..., q_m\}$, output a span $S = \{c_i, c_{i+1}, ..., c_{i+j}\}$ from the original paragraph C. Therefore, the objective function can be define as:

$$L(\theta) = -\frac{1}{N}\sum_{i}^{N}[log(p^1_{y^1_i}) + log(p^2_{y^2_i})]$$

Where y^1_i, y^2_i are the starting and ending positions of the groudtruth of the i-th sample respectively.

2.2 Model overview

Our Ch-QANet is improved on the basis of QANet[13]. Its encoder consists only of convolutional and self-attention[10], where convolutional and self-attention responsible for local and global feature extraction respectively. The same encoder block in right of figure1 is used throughout the model, only varying the number of convolutional layers for each block. Layer norm[1] and residual connection[4] is used between every layer in the encoder block. A positional encoding[12] is added to the beginning of each encoder layer, which consisting of sin and cos functions at varying wavelengths , can makes it easier for attention mechanisms to learn better. Each sub-layer after the positional encoding (one of convolution, self-attention, or feed-forward-net) inside the encoder structure is wrapped inside a residual block. Due to the essential difference between Chinese data sets and English data sets, fewer characters make up words. Therefore, we think it is very difficult for the model to learn the spatial distribution of word vectors in the embedding layer, so we added an extra multi-head attention[10] to the decoder of the model to further capture the correspondence between the question and the paragraph pairs.

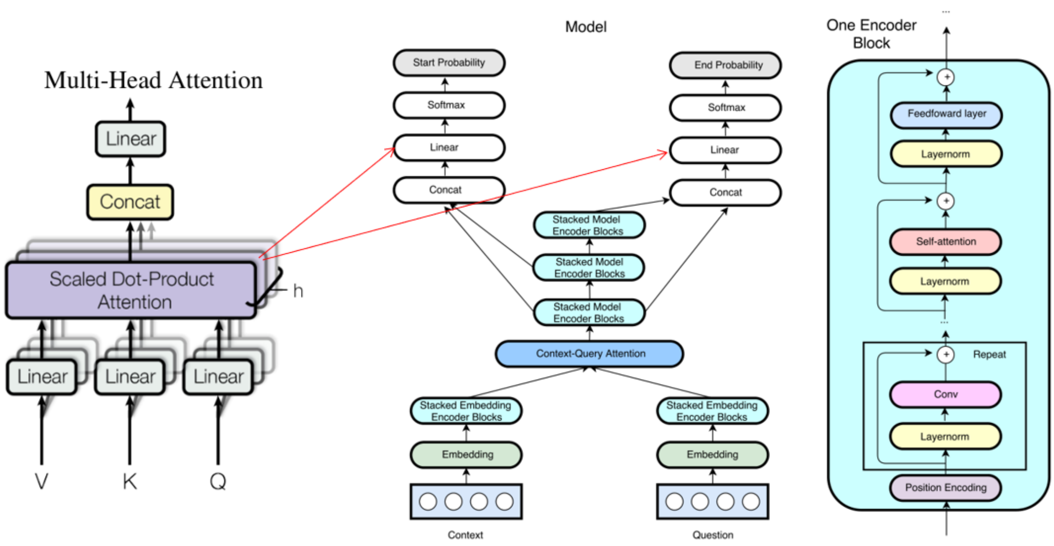

Figure 1: Structure of Ch-QANet

There are three key characteristics in Ch-QANet:

- Using more residual block structure to deepen network depth
- Using extra self-attention ,multi-head attention, to better capture the correspondence between the question and the paragraph pairs
- Enhanced location information by deepen convolution block

Input embedding layer contain word embedding and character embedding. Word embedding can load from pre-trained word vectors, each word vector dimension is p_1 (300 in original QANet, 150 in ours). Denote the word vector corresponding to the word w is x_w. Character embedding is randomly initialized, whose dimension is p_2 (200 in original QANet, 32 in ours), padding or truncating the length of each word to k, then the word w can also be presented as a matrix of $p_2 * k$. After convolution and max-pooling, we get a p_2-dimensional character-level word vector, denoted as x_c. Stitch x_w and x_c to get the word vector of w: $[x_w; x_c] \in R^{p_1+p_2}$. Finally, the word vector will go through a two highway network[8]. What is worth thinking about is that in the character embedding part, the effect of embedding is not good because the Chinese vocabulary consists of fewer words

Embedding Encoder Layer consists of encoder block, which contains position encoding[12], convolution layer, self-attention layer[10] and feedward layers. Every operation will put into the residual block[4]. Convolution captures contextual local structures, while self-attention captures global interactions between texts. Depthwise separable convolutions[2] using there need less parameters than normal convolutions, have stronger generalization ability.

Context-Query Attention Layer is used to find corresponding attention matrix in sematic space of paragraphs and questions. We use $C \in R^{d*n}$ and $Q \in R^{d*n}$ to denote the encoded context and query. The context-to-query attention matrix and the query-to-context attention matrix are calculated according to C and Q. First we calculate the similarity between context and query, and the result matrix is denoted as $S \in R^{d*m}$. The formula of calculating similarity is $f(q, c) = W_0[q, c, q \bigodot c]$, q, c are the intermediate representations of individual words, W_0 is a trainable parameter and $\bigodot$ is element-wise dot. Secondly, we use softmax to normalize the rows and columns of S and get $\overline{S}$ and $\overline{\overline{S}}$. Then the context-to-query attention matrix is $A = \overline{S} \cdot Q^T \in R^{n*d}$, the query-to-context attention matrix is $B = \overline{S} \cdot \overline{\overline{S}}^T \cdot C^T \in R^{n*d}$

Model Encoder Layer consists of three model encoder blocks in right of figure1, each model encoder block is composed of 7 encoder blocks in our Ch-QANet, and parameters are shared between the three model encoder blocks. Similar to BIDAF[7], every input in the position is $[c, a, c \bigodot a, c \bigodot b]$. a and b are the rows of the attention matrix A and B, respectively, and c is the middle representation of the word corresponding to the context.

Output layer predict the probability that each position in the context are the start and end point of the answer span, denoted as p^1, p^2, calculated as follows:

$$p^1 = softmax(W_1[M_0; M_1]); \quad p^2 = softmax(W_2[M_0; M_2])$$

Among them, W_1, W_2 two training variables, M_0, M_1, M_2 in turn correspond to the output of the three model encoder blocks in figure1 (from bottom to top). As shown in the left half of the figure1, we added an extra multi-head self attention structure[10] behind the linear layer to increase the inference capability of the model.

3 Experiments

3.1 Dataset overview

Due to the capability of the computer, we only selected 40,000 records in Baidu zhidao and search as the training set (about 24%), 20000 records as the evalidation set, and 200 records as the prediction samples to see the prediction results (we can see from figure4). You can see $demo.ipynb$ in Appendix for details.

3.2 Evaluation metric

The evaluation metric in this paper are **BLEU-4** and **Rouge-L**. We will use these methods to compare our Ch-QANet with BiDAF[7].

3.3 Implementation details

The basic templates of our project are Baidu's BiDAF[7] code. We added the character dictionary part to the project's $Dataloader$ module because Ch-QANet needs use the word and character embedding separately.

As for the model part, we independently implemented the Ch-QANet. We used pre-trained word embedding in experiment (Baidu Encyclopedia Word2Vec) and found that the model using pretrained embedding has better performance than Baidu's baseline of BiDAF[7].

Implementation of additional multi-head attention in left of figure1 is somehow complicated. We can not repeat single attention block a few times directly because tensorFlow is not automatically paralleled, we need to mask the sequence to ignore the effect of padding. Rather than settig the padding part to zero like traditional mask, the mask in multi-head attention is to subtract a very large integer before soft max, therefore the results will be very close to zero after soft max. (Full code can see in Appendix)

3.4 Individual component analysis

The most important part of our model is the additional attention mechanism in decoder. We added it because QANet had a very bad result at the beginning of our experiment. We guess it was because there were not enough characters per word to build high-dimentional embedding in the Chinese dataset. So we added this section to improve the model's ability to find out the dependency relationship between output answers and text segments. Experiments show that our improvement can improve the evaluation index of the model by about 12 %.

3.5 Performance Comparison

Compared with the time dimension, the speed of Ch-QANet was just twice that of BiDAF[7] (we can see from figure3), partly because we added an extra attention mechanism and use different batchsize (also perhaps in some programing parts, it was not as good as the original paper).

Table 1: Comparison of metric

Dataset	Baidu Search		Baidu Zhidao	
Model	BLEU-4%	ROUGE%	BLEU-4%	ROUGE%
BiDAF non-embedding	21.6	26.2	37.3	41.2
QANet	17.6	21.3	35.4	38.0
Ch-QANet non-embedding	19.7	24.3	37.9	39.8
Ch-QANet with embedding	**24.8**	29.1	**46.1**	**52.9**
Baidu's baseline	23.1	**31.1**	42.2	47.5
Human	55.1	54.4	57.1	60.7

On the comparison of evaluation indicators, we can see from Table 1 that QANet[13] was not better than BIDAF[7] before using pre-trained embedding. After we have made some improvements to the original model (Ch-QANet, deeper conv blocks and additional self-attention),the effect of the model has been improved to a higher level but also worse than BiDAF[7]. This shows that the characteristics of Chinese datasets have a great impact on the embedding structure of QANet[13], so we used pre-training embedding and were surprised to find that it even better than Baidu's baseline though training on subsets. This shows that the particularity of Chinese data sets limits the performance of QANet[13].

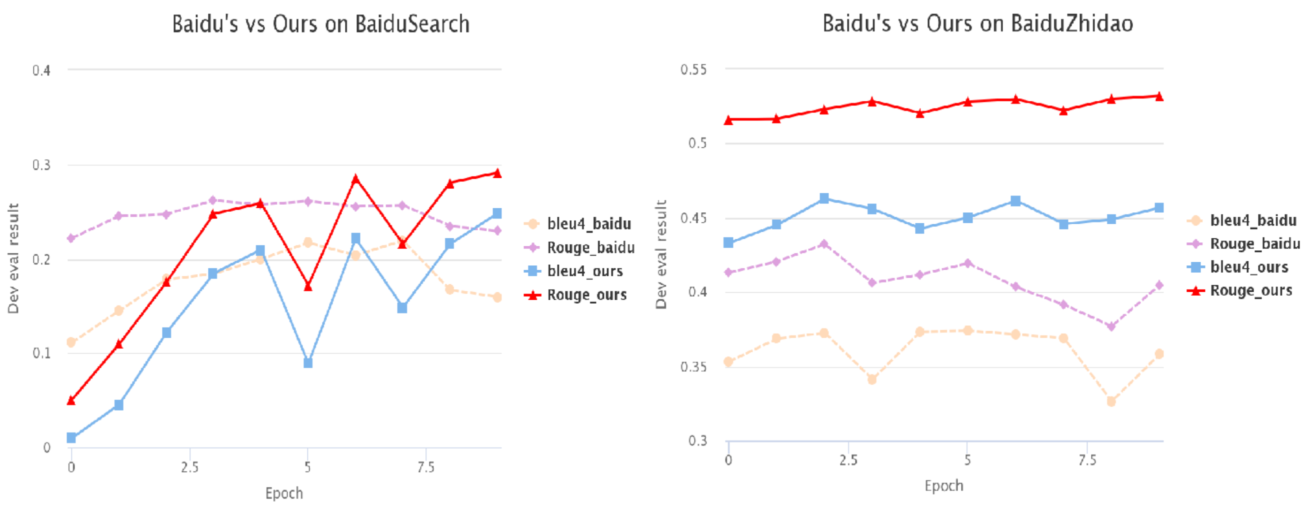

Figure 2: Metric comparision of Ch-QANet and BiDAF

See from figure2, we can also find that the performance of the two models in Baidu search is much lower than that of Baidu zhidao, because the answers in Baidu search are more casual and the generator was not concerned about and modified some inappropriate answers. Therefore, the effect of the model can be better improved in data preprocessing in future work (see in Appendix).

References

[1] K. J. R. . H. G. E. Ba, J. L. Layer normalization. *arXiv preprint*, 2016.
[2] K. J. R. . H. G. E. C. F. Ba, J. L. Xception: Deep learning with depthwise separable convolutions. *arXiv preprint*, 2017.
[3] L. K. L. J. L. Y. Z. S. X. X. . . L. X. He, W. Dureader: a chinese machine reading comprehension dataset from real-world applications. *arXiv preprint*, 2017.
[4] Z. X. R. S. . S. J. He, K. Deep residual learning for image recognition. in proceedings of the ieee conference on computer vision and pattern recognition. *pp. 770-778*, 2016.
[5] R. M. S. X. G. J. T. S. M. R. . D. L. Nguyen, T. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint*, 2016.
[6] Z. J. L. K. . L. P. Rajpurkar, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint*, 2016.
[7] K. A. F. A. . H. H. Seo, M. Bidirectional attention flow for machine comprehension. In *arXiv preprint*, 2018.
[8] G. K. . S. J. Srivastava, R. K. Highway networks. *arXiv preprint*, 2015.
[9] W. T. Y. X. H. J. S. A. B. P. . S. K. Trischler, A. Newsqa: A machine comprehension dataset. *arXiv preprint*, 2016.
[10] S. N. P. N. U. J. J. L. G. A. N. . . P. I. Vaswani, A. Attention is all you need. in advances in neural information processing systems. *pp. 5998-6008*, 2017.
[11] . J. J. Wang, S. Machine comprehension using match-lstm and answer pointer. *arXiv preprint*, 2016.
[12] Z. V. . S. R. Xiong, C. Dynamic coattention networks for question answering. *arXiv preprint*, 2016.
[13] D. D. L. M. T. Z. R. C. K. N. M. . L. Q. V. Yu, A. W. QANet: combining local convolution with global self-attention for reading comprehension. In *arXiv preprint*, 2018.

Appendix

1. Full experiment results

(1) Time comparision

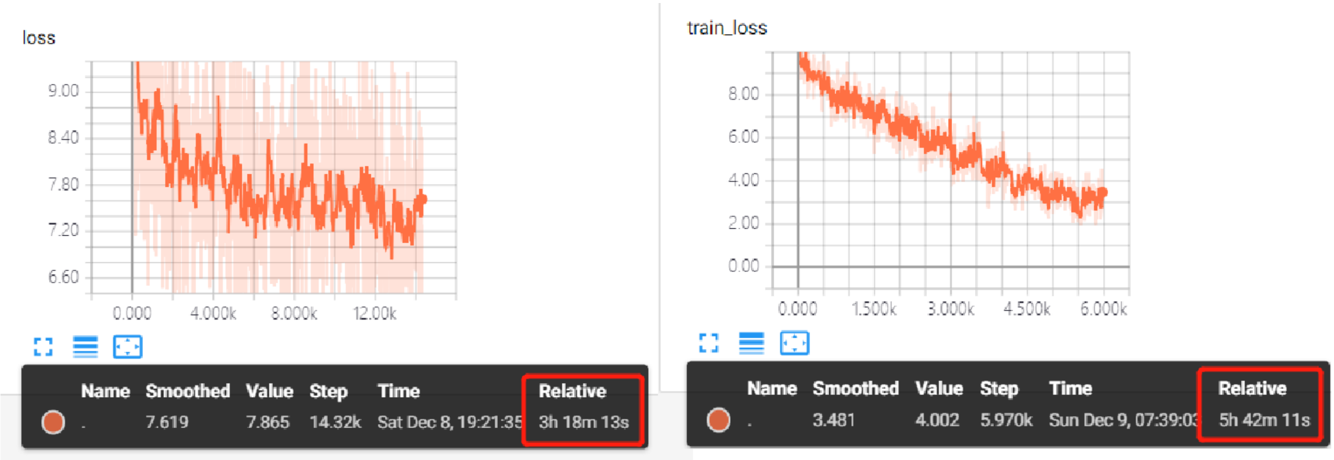

Figure 3: Time comparision of Ch-QANet and BiDAF

(2) Predict results

Figure 4: Predict results of Ch-QANet

(3) More experiment results can see from *Code and usage*.

2. Acknowledgement

Thanks for Pro Li and TAs' patient teaching in this semester, we have successfully entered the hall of in deep learning. Thankd for idea provider Jinghong Lin Wei Qing and Yutao Liao.

3. Code and usage

1.Download code from:
https://github.com/yanx27/DuReader_QANet_BiDAF

2.Download dataset, pre-trained models, experimental data and results (BaiduNetDisk):
https://pan.baidu.com/s/1qoxnF00wyJ2dqcAPDYTb8w password:gn5b

Download the data file of BaiduNetDisk and replace the data file in Github code.

3.Run the model (Download the required library by yourself, tensorflow>=1.9, jieba, etc.):
(1) Baidu's model - BiDAF
Building a dictionary: $pythonBaiduRun.py - -prepare$
Training: $pythonBaiduRun.py - -train$
Validation: $pythonBaiduRun.py - -evaluate$
Test: $pythonBaiduRun.py - -predict$

(2) Our model - Ch-QANet
Building a dictionary: $pythonOurRun.py - -prepare$
Training: $pythonOurRun.py - -train$
Validation: $pythonOurRun.py - -evaluate$
Test: $pythonOurRun.py - -predict$

(If you just validate the experiment results, you can validate it directly. Training and building a dictionary will cover the original trained model!)

4. $./demo$ folder holds a small example of the model that is easy for you to use and understand.

5. $./data/demo$ folder stores the experimental data, including respectively two 20,000 training sets, two 10,000 validation sets and two 100 test sets (only used to test output predictions) of Baidu zhidao and Baidu search.

6. $./data/model$ folder stores the training results of two models respectively with 10 epoch in two data sets, which may be slightly higher or lower than the experimental results (since the experimental result is the best indicator of 10 epoch, which are documented at ./data/summary and tensorboard).

7. $./data/result$ folder stores the output of the answer.

8. $./data/summary$ folder stores the log of loss value of experiment and the tensorboard, which can be opened with $tensorboard - -logdir = thispath$.

9. $./data/vocab$ folder stores the trained dictionary, which is the result of the running of $python\ xx.py - -prepare$.

10. Thanks for Prof.Li and TAs' hard working. If there is any problem with the code, please contact me in time.

4. Future work

In the future work, there are the following improvements:

(1) **Data augmentation:** Each question has multiple artificial reference answers. The baseline system only uses the first reference answer to generate a piece of data. The model uses the text interval position as the representation, but the reference answer is not necessarily. In the paragraph, it is easy to produce wrong training materials.

(2) **Make good use of all resources:** Amplify multiple reference answers as a training data set. Maintain training data quality and filter out training data with a similarity score below 0.7. The discovery system selects the sentence and the retelling question as the answer, and preprocesses the most relevant paragraphs for prediction. Then using the method of full-text concatenation, the paragraphs of the article are connected in series, and the whole article is used to predict the answer.

(3) **Separate modeling** of Yes or No questions and entity questions.