Building a QA system (IID SQuAD track)

I-Chien Lai[1], Pei-Wei Kao[1], Ta-Hsuan Chao[1]

[1]Department of Electrical Engineering, Stanford University

Stanford University

Problem

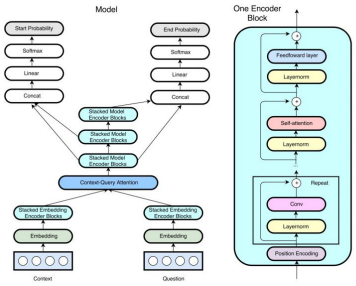
The advent of artificial intelligence and machine learning have greatly eased many tasks of our daily lives. In this final project, we focus on one very interesting task, which is reading comprehension. The task is difficult because even humans have difficulty doing reading comprehension asks. One has to fully understand the passage and the question in order to get the right answer, thus the task would even more difficult to teach machines how to answer reading comprehension problems. Thus, the problem we are trying to solve is Question Answering on the SQuAD dataset.

Background

Our goal of this project is to implement a question answering system that works well on SQuAD 2.0 by capturing long-term dependency. As we believe that Transformer-based models is a trending structure in recent years, we would like to implement a Transformer-based model on QA task, QANet, to see how it improve the performance on SQuAD 2.0, compared to our baseline model BiDAF. To achieve better performance on longer contexts, we want to adapt the ideas from Transformer-XL to QANet to see if it could learn longer-term dependencies of the texts.

Methods

We choose to implement QANet, a Transformer-based model that combines local convolution with global self-attention.



In Encoder layer, we adapted Transformer-XL to our QANet model with the following improvements.

- Recurrence Mechanism
- Relative Positional Encoding

Analysis

We compare the following components in the QANet model:

- Embedding: Word vs. Word + Character embedding
- Self Attention: Single-head vs. Multi-head attention
- Layer Dropout: With vs. Without layer dropout trick
- Feed Forward Layer: Convolution vs. Linear layer

	Embedding	Layer Dropout	Attention	Feed Forward	EM	F1	Difference
QANet	word + char	✓	single-head	convolution	62.83	66.4	
	word	✓	single-head	convolution	60.46	64.19	-2.37 / -2.21
	word+char	✗	single-head	convolution	64.85	68.48	**2.02 / 2.08**
	word+char	✓	single-head	linear	61.35	64.86	-1.48 / -1.54
	word+char	✗	single-head	linear	60.61	64.04	-2.22 / -2.36
	word+char	✓	multi-head	convolution	64.19	67.68	**1.36 / 1.28**
	word	✓	multi-head	convolution	62.39	65.77	-0.44 / -0.63

Our findings:

- Using character embedding plus word embedding achieved better score on validation set than only using word embedding
- Multi-headed attention could generalize the token-token interaction better than single-headed attention
- Discarding the layer dropout trick and using constant dropout in the EncoderBlock surprisingly improved the score and achieved a comparable score with the multi-headed attention model
- Using convolution layer as feed forward layer in both the EncoderBlock and output module yielded higher results than using linear layer, which shows that convolution layer could better capture the local structure of the context

Experiments

- **Data**: We used the SQuAD 2.0 provided by the starter repository.
  - train: 129,941 examples
  - dev: 6078 examples
  - test: 5921 examples
- **Evaluation method**:
  - F1: Measure the portion of overlap tokens between the answer predicted by our model and the ground truth.
  - EM: 1 if the prediction of our model is exactly the same as ground truth, and 0 otherwise. This metric is stricter than that of the F1 score.
- **Experimental details**: For experiment, we implemented three different models on this task and all reached good results, including BiDAF as the baseline, QANet, and QANet-XL. We trained these models on Azure's virtual machine (Standard NC6 v3), with the following settings.
  - dropout rate: 0.1
  - number of head: 8
  - hidden size: 128
  - memory length: 256
- **Results**: The following table showed that our QANet has successfully exceeded the baseline model in both the EM and F1 score. On the other hand, QANet-XL should improve due to the recurrence mechanism and the relative position encoding, but the result showed that QANet-XL didn't achieve a higher score compared to the original QANet. We think it's because the paragraphs in SQuAD dataset are not that long, they don't need rely on Transformer-XL to achieve a remarkable result.

dev	EM	F1
BiDAF	62.141	65.606
QANet	**64.191**	**67.677**
QANet-XL	60.561	63.430

test	EM	F1
BiDAF	60.575	64.235
QANet	**60.727**	**64.087**
QANet-XL	57.853	61.022

Conclusions

We believe this is a very interesting final project. After this work, we not only dealt with a very special task, Question Answering, but also learnt a lot of interesting methods to solve this question. We successfully implemented the Question Answering task along with several baselines and ablation studies, and attained a great deal of natural processing knowledge in this process.

In this project, we concluded that transformer-based model, namely, the QANet, performed better than RNN-based model, namely, the baseline BiDAF. In addition to that, we concluded that adding character embeddings would also greatly improve the performance of the model compared to that with only word embeddings. However, we also found out that adding the to tricks of the Transformer-XL into our model would not improve the performance. Furthermore, according to our ablation study, we found out that the convolution layers of the QANet is truly beneficial to our model. However, the dropout trick of the QANet isn't really beneficial to the model in this dataset, and cancelling out the dropout trick would even yield higher results. Thus, the highest model we can achieve is using the QANet with character and word embeddings, but cancelling out the original dropout trick.

This conclusion is not surprising because each dataset has its own unique method to achieve highest performance. We must find the most suitable method according to the characteristic of the dataset, and we also have to deal with overfitting problems. Thus, this concludes what we learnt in the final project, and we believe what we discovered is worth sharing for all who are interested in the Question Answering task.

References

[1] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension, 2018.

[2] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension, 2018.

[3] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019.

Acknowledgments

First, would would like to thank the course 224n to give a chance to learn about the interesting concepts of natural language processing, and the chance to do a final project together on such an interesting problem. We would also like to thank the instructor for his diligent teaching, and our mentor for her informative feedbacks on our proposal and milestone..