Adapting Transformer-XL to QANet for SQuAD 2.0

Lorraine Zhang {lz2017}@stanford.edu

Motivation

- Explore a novel approach to reading comprehension system
- Experiment with deep learning techniques for question answering
- Improve QANet performance on SQuAD 2.0

Data

- **Source**: https://github.com/chrischute/squad.git
- **Datasets** = {Training set: 129,941 examples, Dev set: 6078 examples, Test set: 5915 examples}
- **Pretrained GloVectors**: 300-dimensional embeddings trained on CommonCrawl 840B corpus.

Results

Models	F1	EM	Epochs
Baseline BiDAF	61	57.45	30
Baseline QANet, SQ1.0	76.2	66.3	30
QANet, dev set	**68.46**	**64.81**	30
QANet, test set	**65.18**	**61.56**	30
QANet+Transformer-XL	64.87	61.75	30

Table 1: F1, EM scores, non-PCE

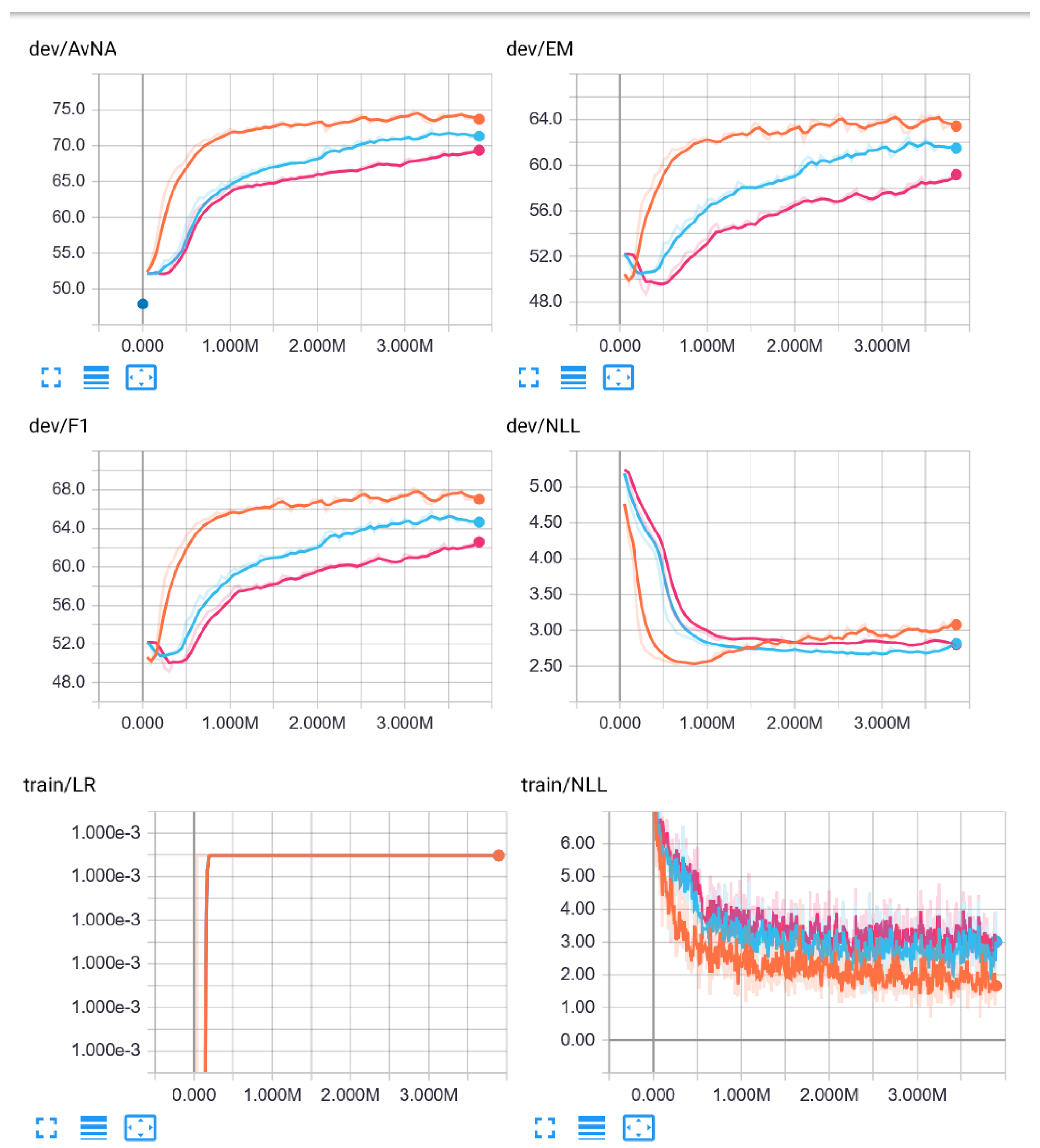


Figure 1: TensorBoard visualization(Orange: QANet, Blue: QANet-XL)

Problem Definition

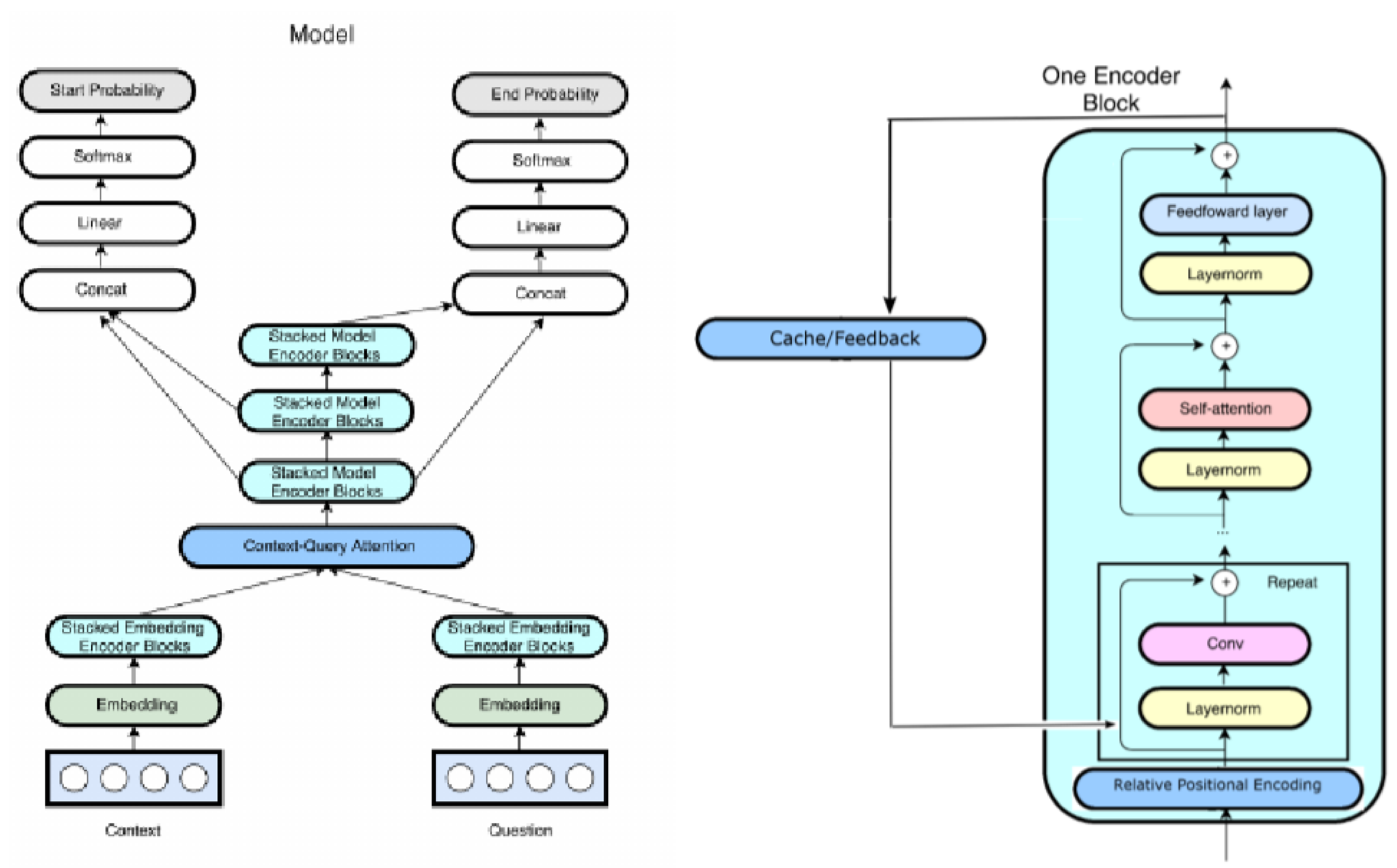
Challenge:

- Answer questions correctly in longer context on a reading comprehension system
- Many models such as QANet are limited by fixed_length dependency

Evaluation Metric:

- **EM** score:
  - Exact Match to ground truth answer
  - Binary measure (true/false)
- **F1** score:
  - Harmonic mean of precision and recall
  - F1 = 2 x prediction x recall / (precision + recall)

Approach



QANet-XL Model:

- **Cache memory and feedback to EncoderBlock**
- **Only use convolution and self attention**
- **Adam optimizer with warm-up rate**
- **Layers:**
  1. Input Embedding Layer
  2. Embedding Encoder Layer
  3. Context-Query Attention Layer
  4. Model Encoder Layer
  5. Output Layer
- **Modified self attention for relative encoding**

Transformer-XL Techniques used:

- **Recurrence Mechanism**
  Cache and reuse hidden states as memory for the current state
- **Relative Positional Encoding Scheme**
  only encode the relative positional information in the hidden states
- **New Variables Introduced:**
  1. mems: previous state
  2. r: relative positional encoding
  3. r_r_bias
  4. r_w_bias

Difference in Self Attention in EncoderBlock:

$$A^{abs}_{i,j} = q_i^T k_j = \underbrace{E_{x_i}^T W_q^T W_k E_{x_j}}_{(a)} + \underbrace{E_{x_i}^T W_q^T W_k U_j}_{(b)} + \underbrace{U_i^T W_q^T W_k E_{x_j}}_{(c)} + \underbrace{U_i^T W_q^T W_k U_j}_{(d)}, \text{ (Transformer)}$$

$$A^{rel}_{i,j} = q_i^T k_j = \underbrace{E_{x_i}^T W_q^T \mathbf{W_{k,E}} E_{x_j}}_{(a)} + \underbrace{E_{x_i}^T W_q^T \mathbf{W_{k,R}} \mathbf{R_{i-j}}}_{(b)} + \underbrace{\mathbf{u^T} \mathbf{W_{k,E}} E_{x_j}}_{(c)} + \underbrace{\mathbf{v^T} \mathbf{W_{k,R}} \mathbf{R_{i-j}}}_{(d)}, \text{ (Transformer XL)}$$

Analysis

- Both QANet and QANet-XL outperformed the baseline BiDAF in F1 and EM scores.
- Recurrence mechanism increased memory requirement on hardware noticeably
- QANet-XL underperformed vanila QANet.
- Limitations on time and hardware prevented adquate training of QANet-XL.
- Had to use different hyperparameters due to lack of available memory
- QANet-XL: 46 hidden size, 4 heads vs QANet: 128 hiddent size, 8 on hidden size
- Lower NLL and steeper trajectory of F1/EM of QANet-XL indicate its promise
- Datasize and character embedding dimension impacted F! and EM scores
- Underperformed original QANet paper whiich used 3x augmented dataset and 200-dimension character embedding vs 96 char-dim in this project

Conclusions

- QANet-XL holds promise to outperform QANet given enough time and resource
- Access to larger dataset and higher performance hardware are essential to meaningful research results

Future work

- Increase dataset size to get better pre-train model
- Increase hidden size and number of heads on higher performance hardware with more memory

References

1. CS224N Default Final Project Handout, Stanford University. February 2019.
2. Adam Wei Yu, David Dohan, . QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension. April 2018.
3. Zihang Dai, Zhilin Yang, Minh-Thang Luong. Transformer-XL: Attentive Language Models Beyond a Fixed-length Context. January 2019.