Question-Answering: QANet++

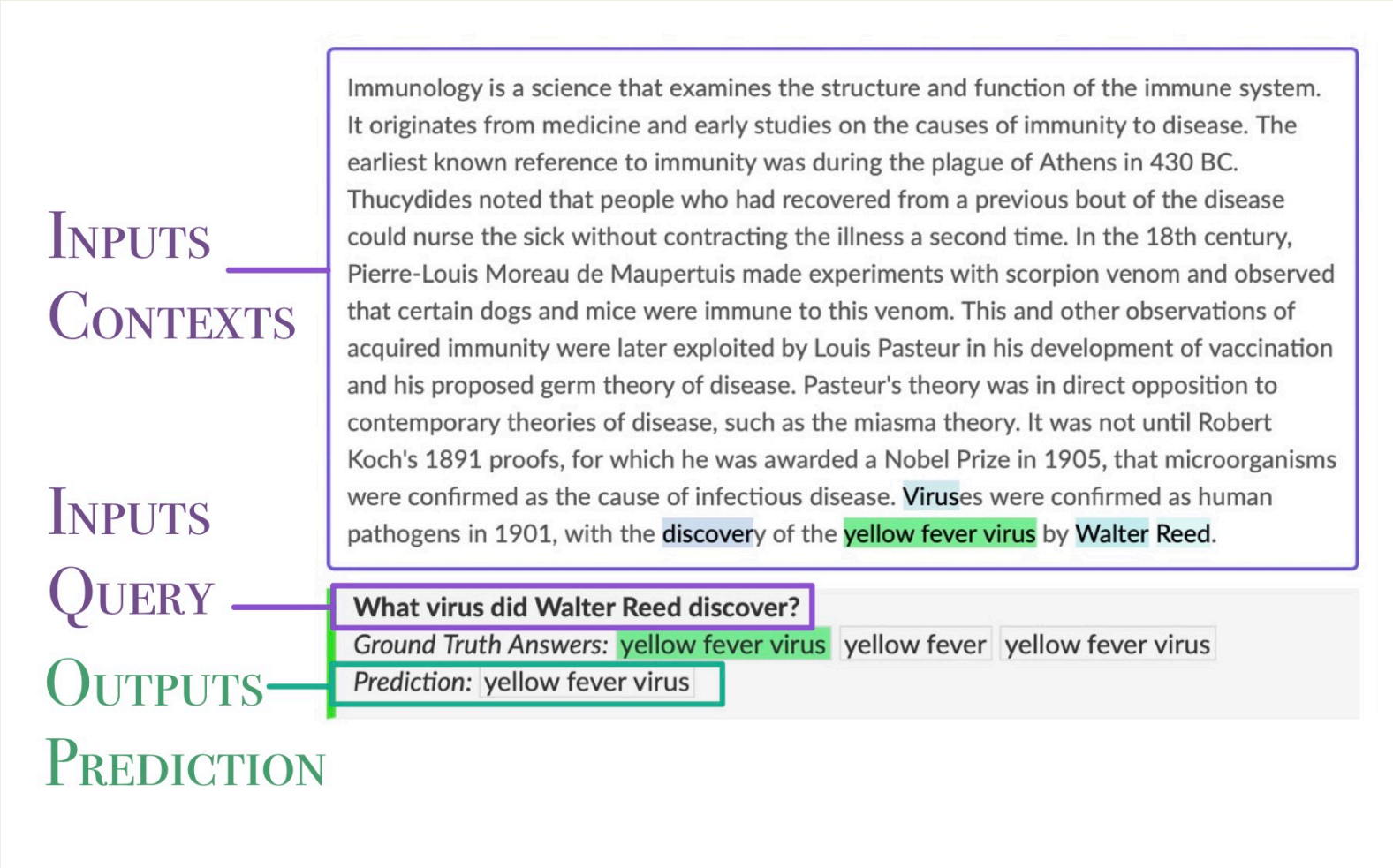
ECE 449 Team 9

Hanyin Shao, Yiqing Du, Huili Tao, Xinkai Yuan, Shiqi Yu

Question Answering & SQuAD

- **Input:** Query and Context
- **Output:** Segment of the context (start index and end index)

Stanford **Qu**estion **A**nswering **D**at aset (SQuAD) is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles.



Evaluation of Model Performance

F1 Score: Measures the **average overlap** between the prediction and ground truth answer.

EM: Measures the percentage of predictions that **exactly match** the ground truth answers

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452

Related Work

- BiDAF
- BERT
- QANet

Context:

Singapore is a small country located in Southeast Asia.

Query:

Where is Singapore situated?

BiDAF's Answer:

Southeast Asia.

BiDAF Bi-Directional Attention Flow

- ***A modular architecture :***

Made out of LEGO blocks with the blocks being "standard" NLP elements

- Closed-domain, extractive Q&A model that can only answer factoid questions

BiDAF Layer Design and Weakness

- Different levels of granularity
- Bi-directional attention flow mechanism to achieve a query-aware context representation without early summarization.

WEAKNESS: Too Slow!

Its recurrent structure cannot be parallel computed, making training process quite slow There had been proposed to avoid using RNNs, but it largely sacrifices the performance.

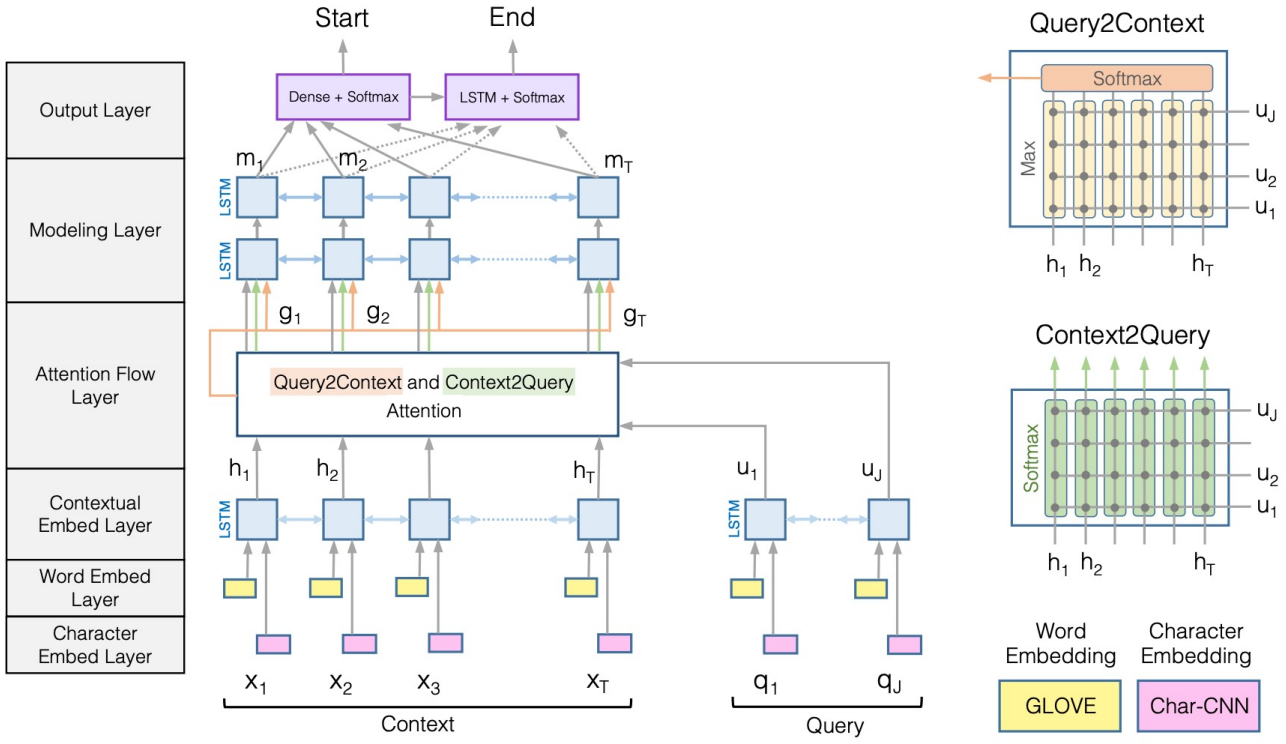
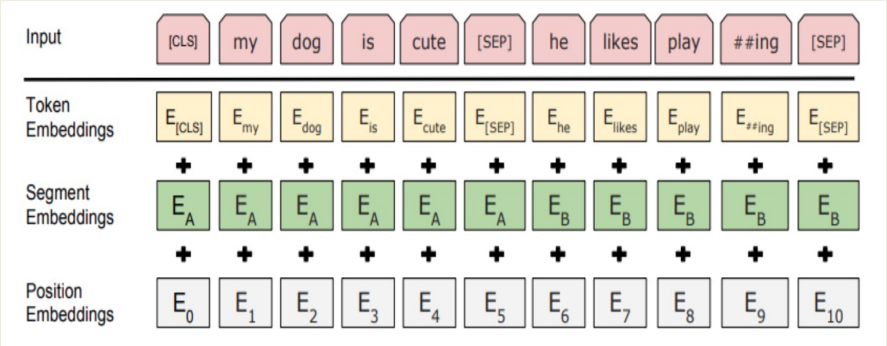
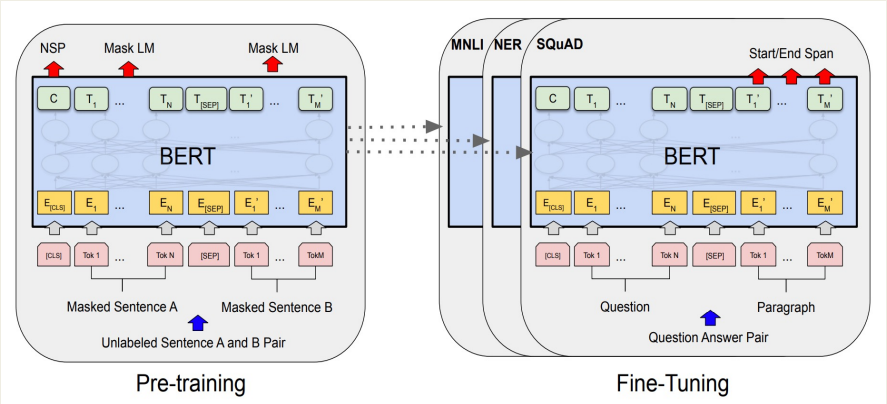
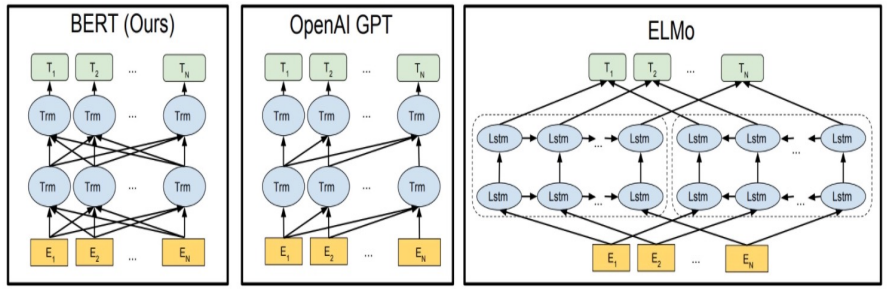


Figure 1: BiDirectional Attention Flow Model *(best viewed in color)*

Embedding Layers → Attention and Modeling Layers → Output Layers







The arrows indicate the information flow from one layer to the next. The green boxes at the top indicate the final contextualized representation of each input word.

BERT- Bidirectional Encoder Representation from Transformers

- *Pre-training + Fine-tuning*
- **BERT's Architecture**

A stack of Transformer's Encoder

NOT use RNNs at all

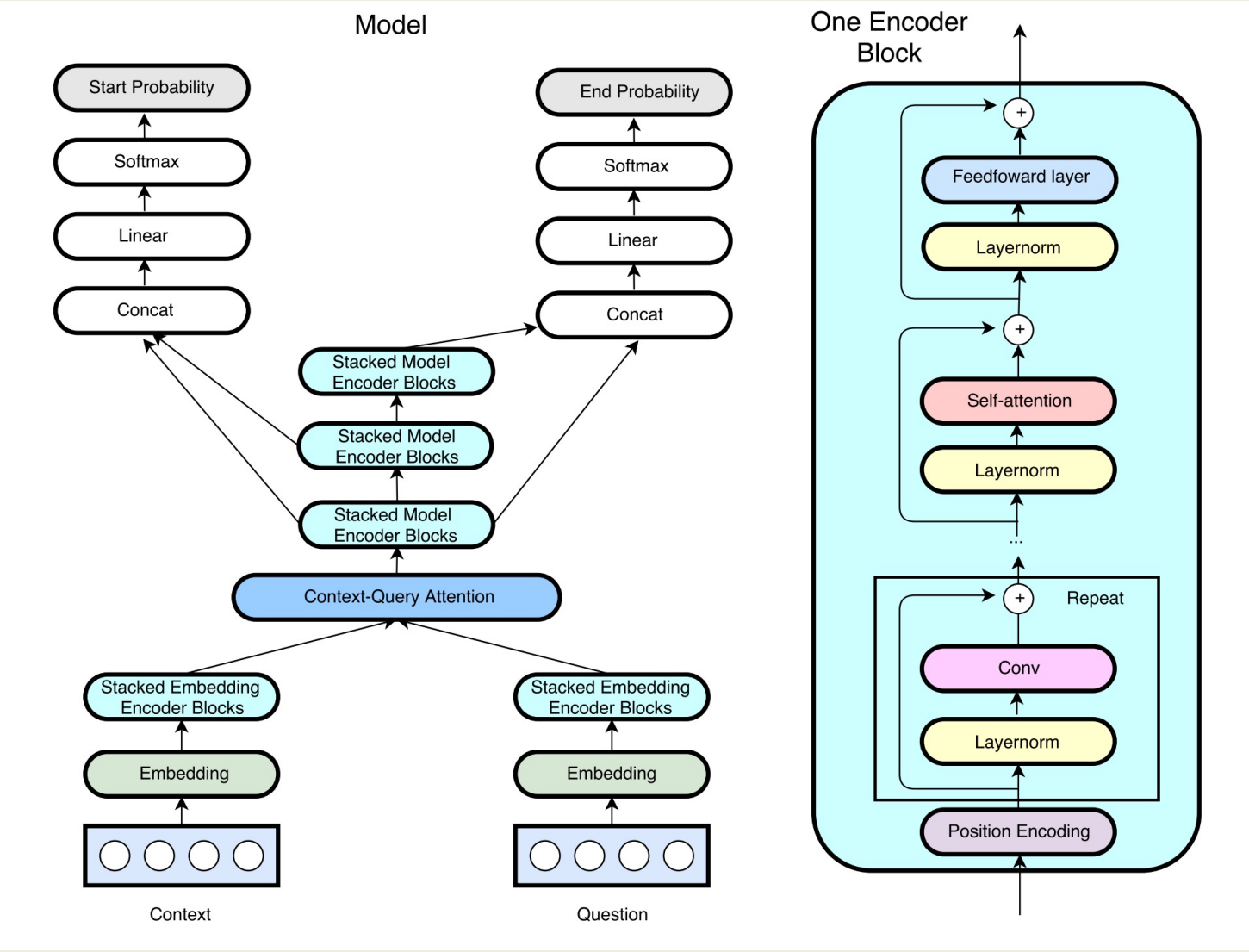
- **Text Processing steps:**

Input representation is constructed by summing the corresponding token, segment, and position embeddings

- **Pre-training Tasks**

Masked Language Modeling +Next Sentence Prediction

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2018.



QANet

- Does not use RNNs at all, and instead use **convolutions and self-attentions** in its feed-forward type architecture
- **Solved problem of BiDAF:** introduces convolutional network which can take advantage of GPU parallel computation.

Why QANet

	QANet	RNN-1-128	Speedup	RNN-2-128	Speedup	RNN-3-128	Speedup
Training	**3.2**	1.1	**2.9x**	0.34	**9.4x**	0.24	**13.3x**
Inference	**8.1**	2.2	**3.7x**	1.3	**6.2x**	0.92	**8.8x**

Table 3: Speed comparison between our model and RNN-based models on SQuAD dataset, all with batch size 32. RNN-x-y indicates an RNN with x layers each containing y hidden units. Here, we use bidirectional LSTM as the RNN. The speed is measured by batches/second, so higher is faster.

	Train time to get 77.0 F1 on Dev set	Train speed	Inference speed
QANet	3 hours	102 samples/s	259 samples/s
BiDAF	15 hours	24 samples/s	37samples/s
Speedup	**5.0x**	**4.3x**	**7.0x**

Table 4: Speed comparison between our model and BiDAF (Seo et al., 2016) on SQuAD dataset.

Data Preprocessing

- **Word embedding:**

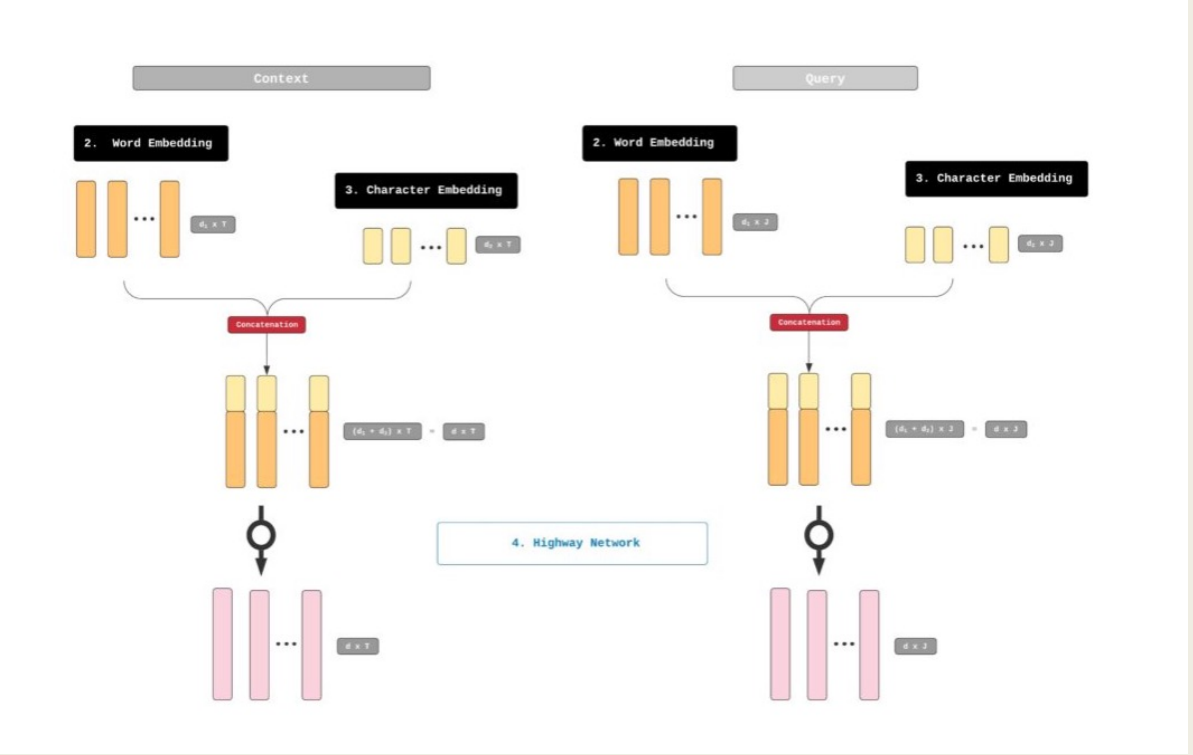
  300 dimensional pre-trained GloVe word vectors (Pennington et al., 2014)

- **Character embedding:**

  Use one-dimensional convolutional neural network (1D-CNN)

- **Highway network:**

  Vertically concatenate previous representations, to adjust the relative contribution from the word embedding and the character embedding steps.



$$\mathbf{z} = \mathbf{t} \odot g(\mathbf{W}_H \mathbf{y} + \mathbf{b}_H) + (\mathbf{1} - \mathbf{t}) \odot \mathbf{y}$$

$\mathbf{W}_H, \mathbf{b}_H$: Affine transformation

$\mathbf{t} = \sigma(\mathbf{W}_T \mathbf{y} + \mathbf{b}_T)$: *transform* gate

$\mathbf{1} - \mathbf{t}$: *carry* gate

QANet

Start Probability

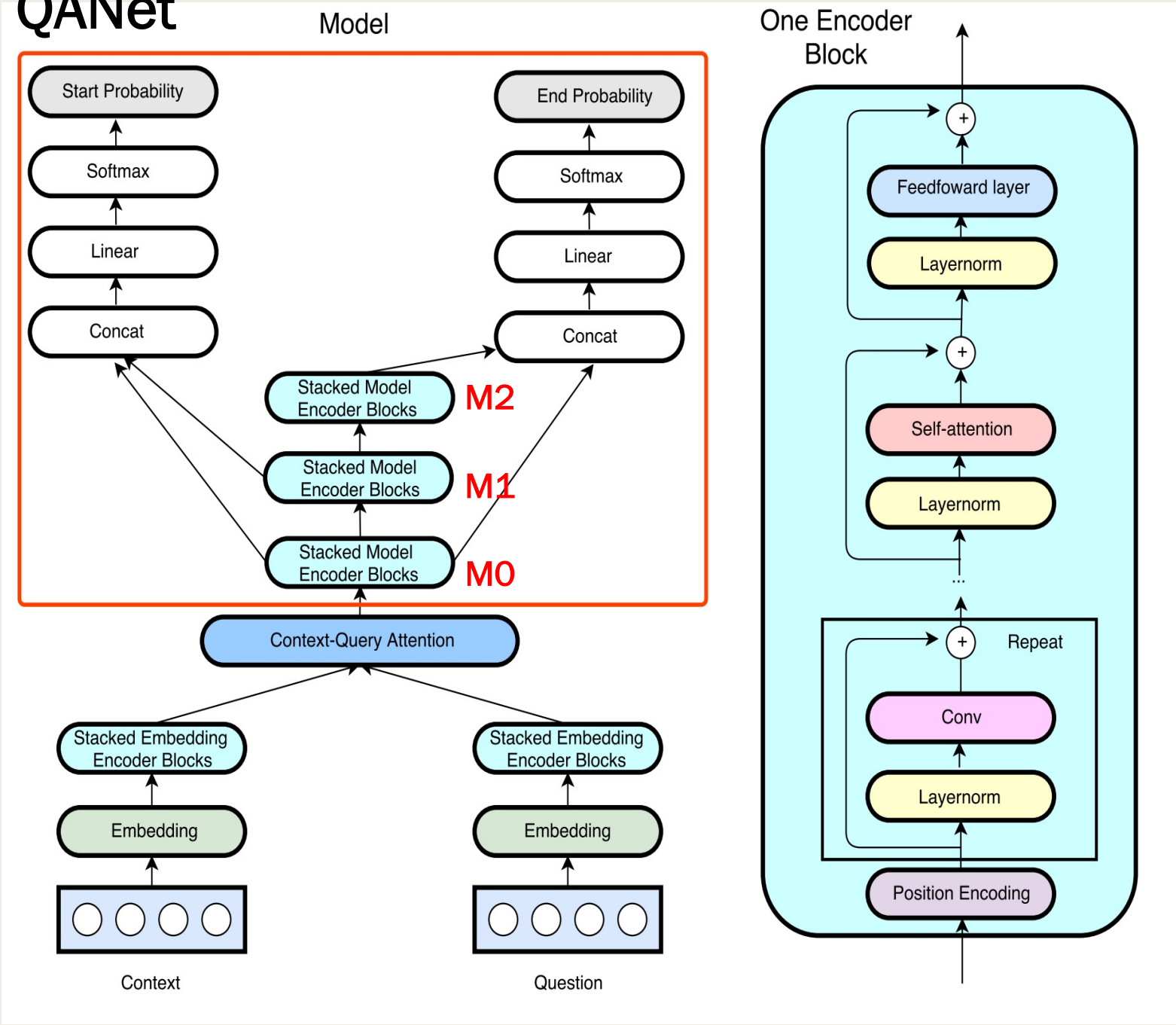
$$p^1 = softmax(W_1[M_0; M_1])$$

End Probability

$$p^2 = softmax(W_2[M_0; M_2])$$

Loss Function

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



Modified QANet 1

QANet+Add

Motivation

- Start probability should be good indicator for end probability

Start Probability

$$p^1 = Softmax\ (W_1[M0; M1])$$

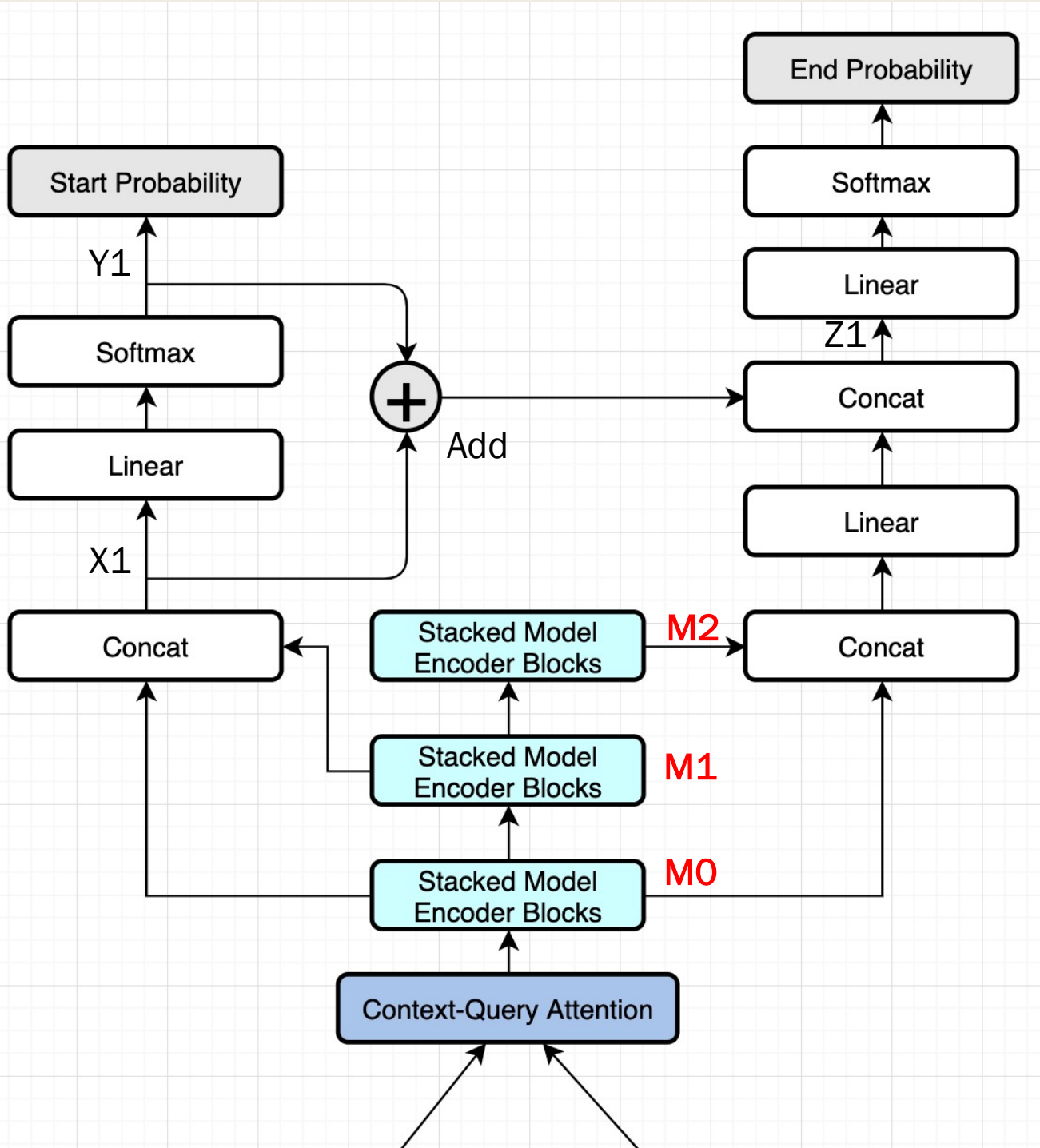
End Probability

$$X1 = [M0; M1]$$

$$Y1 = Softmax(W_1[M0; M1])$$

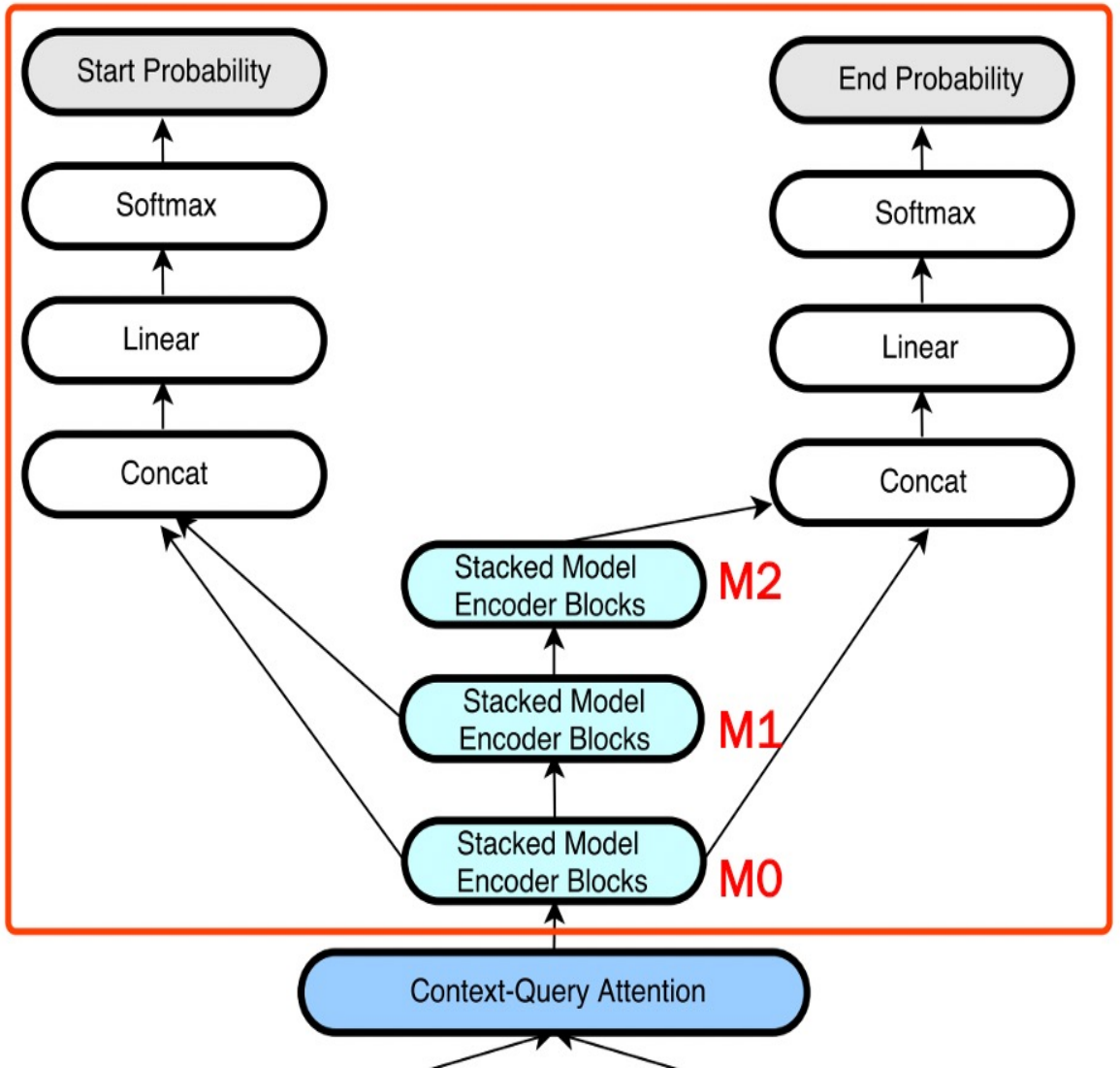
$$Z1 = ((W_2[M0; M2]); [X1 + Y1])$$

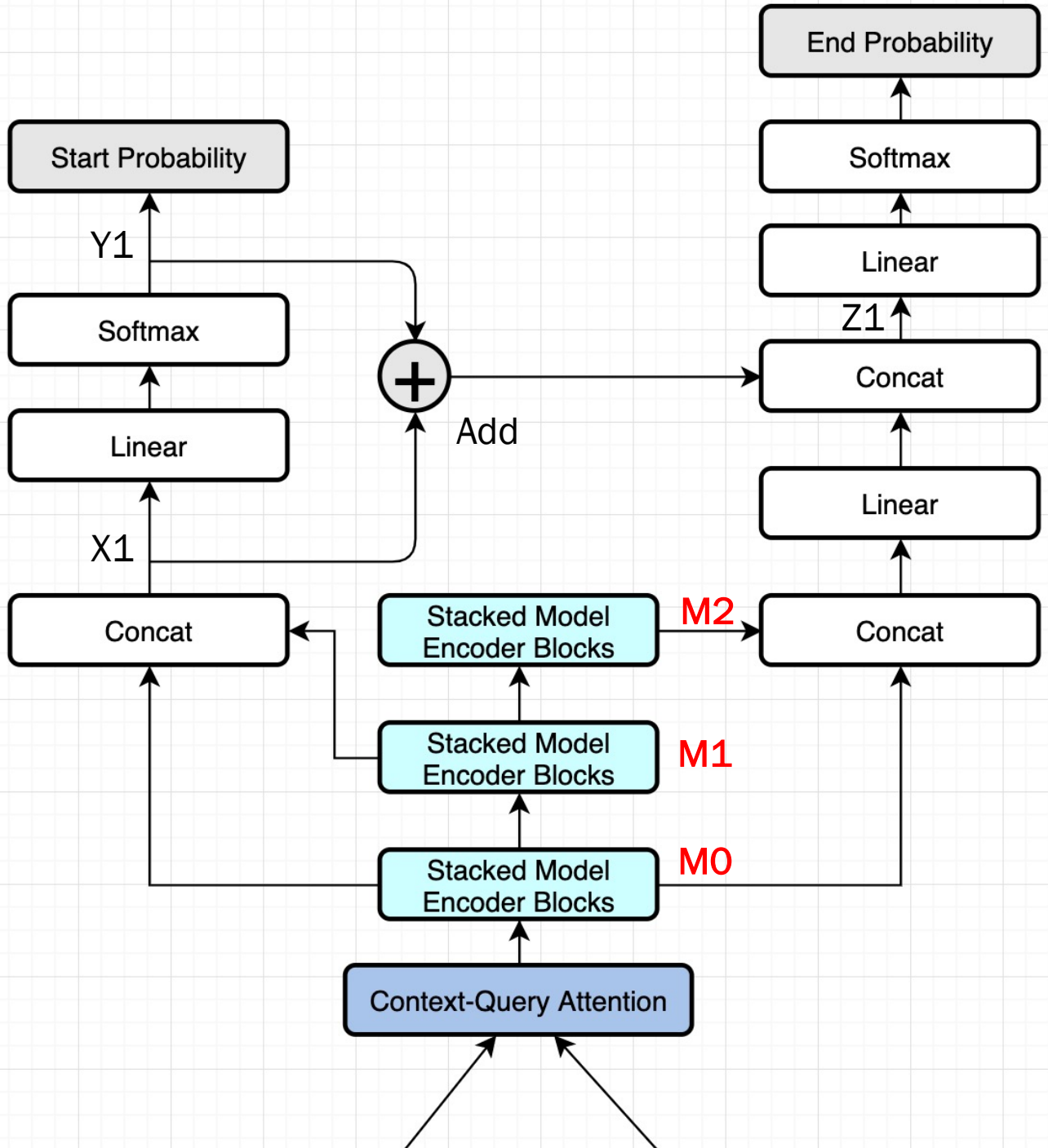
$$p^2 = Softmax\ (W_3 Z1)$$



Modified QANet 1

QANet+Add





Modified QANet 2

QANet+Add+Bias

Motivation

- Maybe linear transformation with bias is able to capture better features
- Bias makes the model more robust

Start Probability

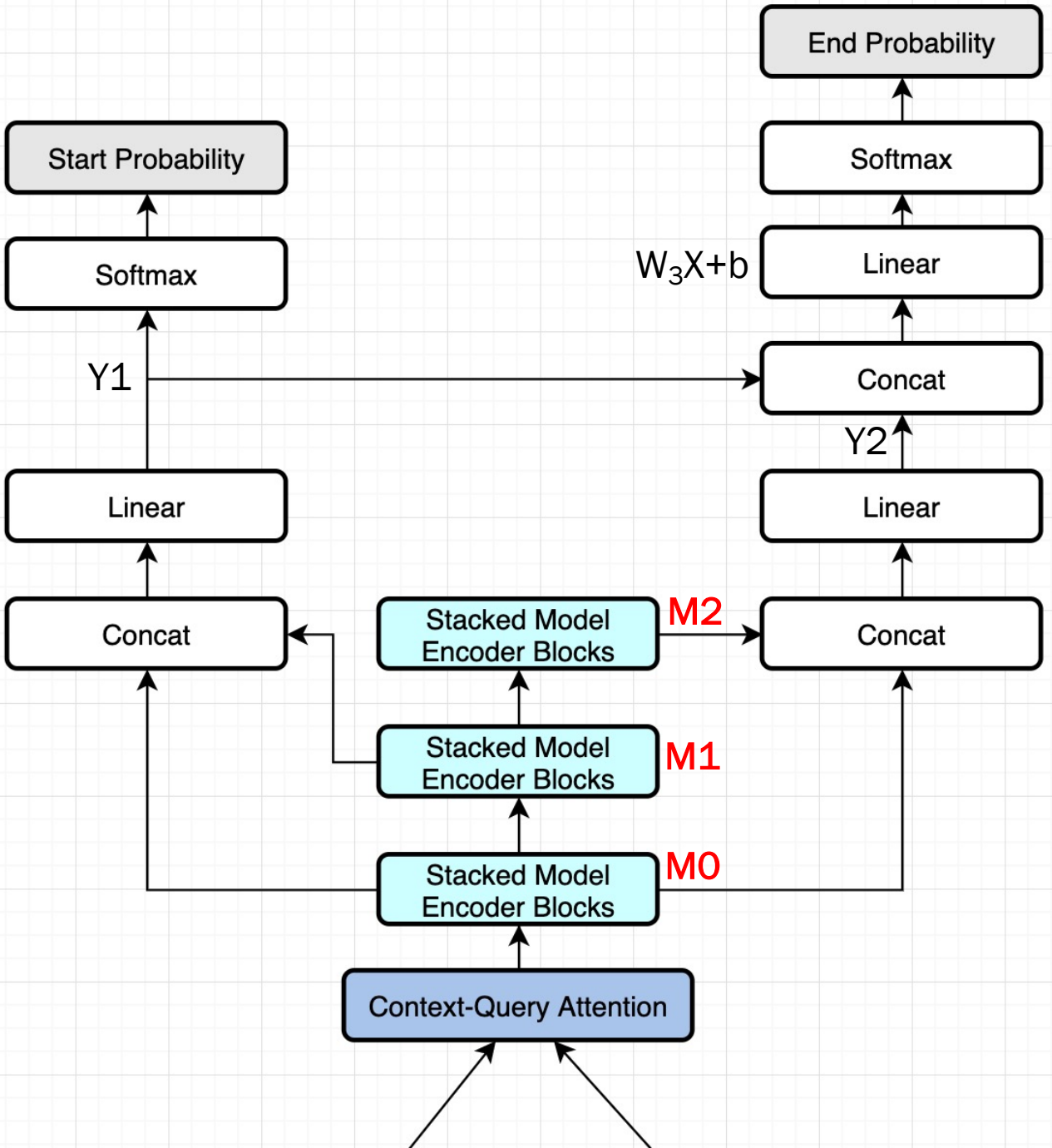
$$p^1 = Softmax\ (W_1[M0; M1])$$

End Probability

$$Y1 = W_1[M0; M1]$$

$$Y2 = W_2[M0; M2]$$

$$p^2 = Softmax\ (W_3[Y1; Y2] + b)$$



Modified QANet 3

QANet+Multiplication+Bias

Motivation

- Contexts around the start have higher probabilities to be the end

Start Probability

$$p^1 = Softmax\ (W_1[M0; M1])$$

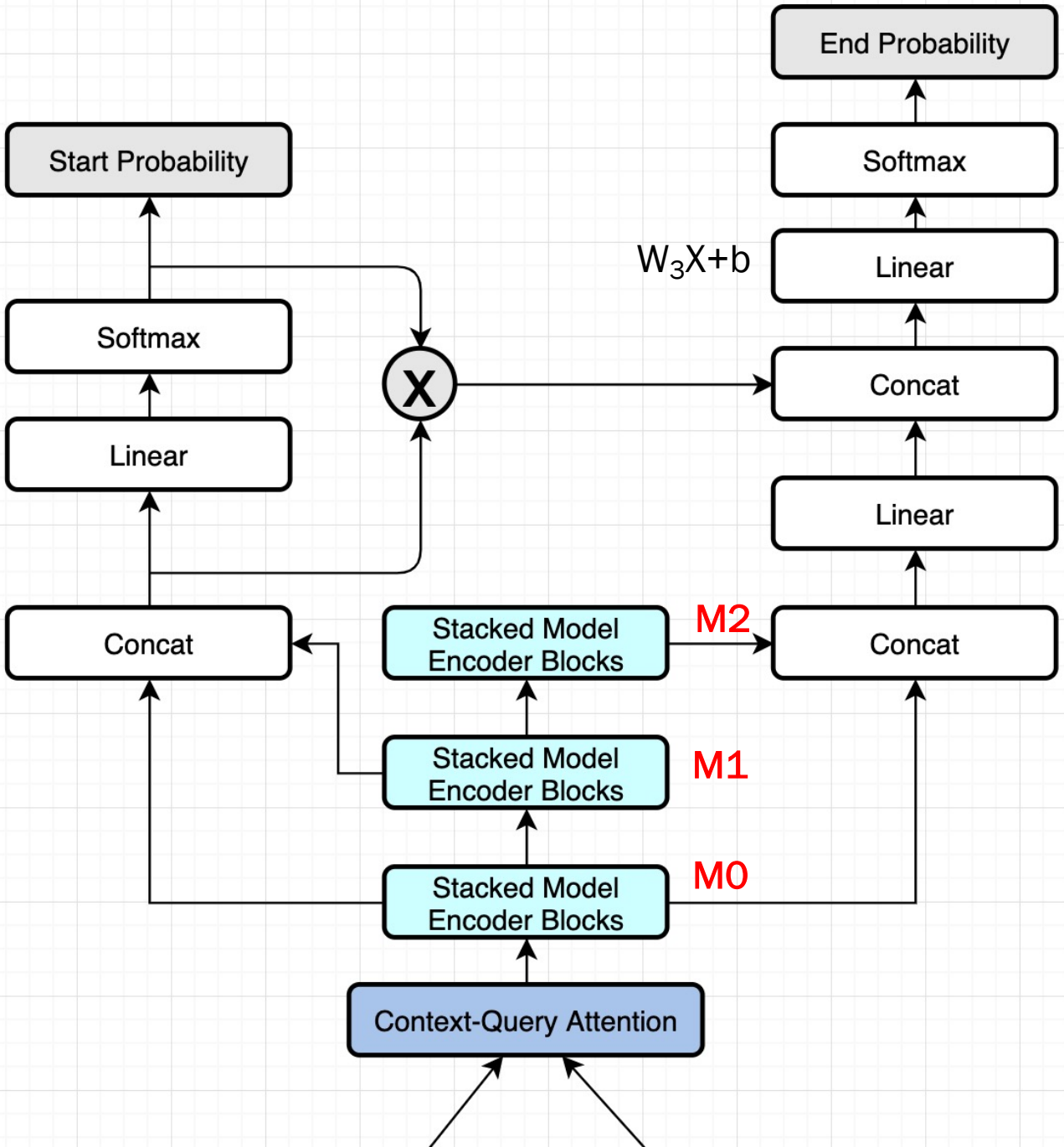
End Probability

$$X1 = [M0; M1]$$

$$Y1 = Softmax(W_1[M0; M1])$$

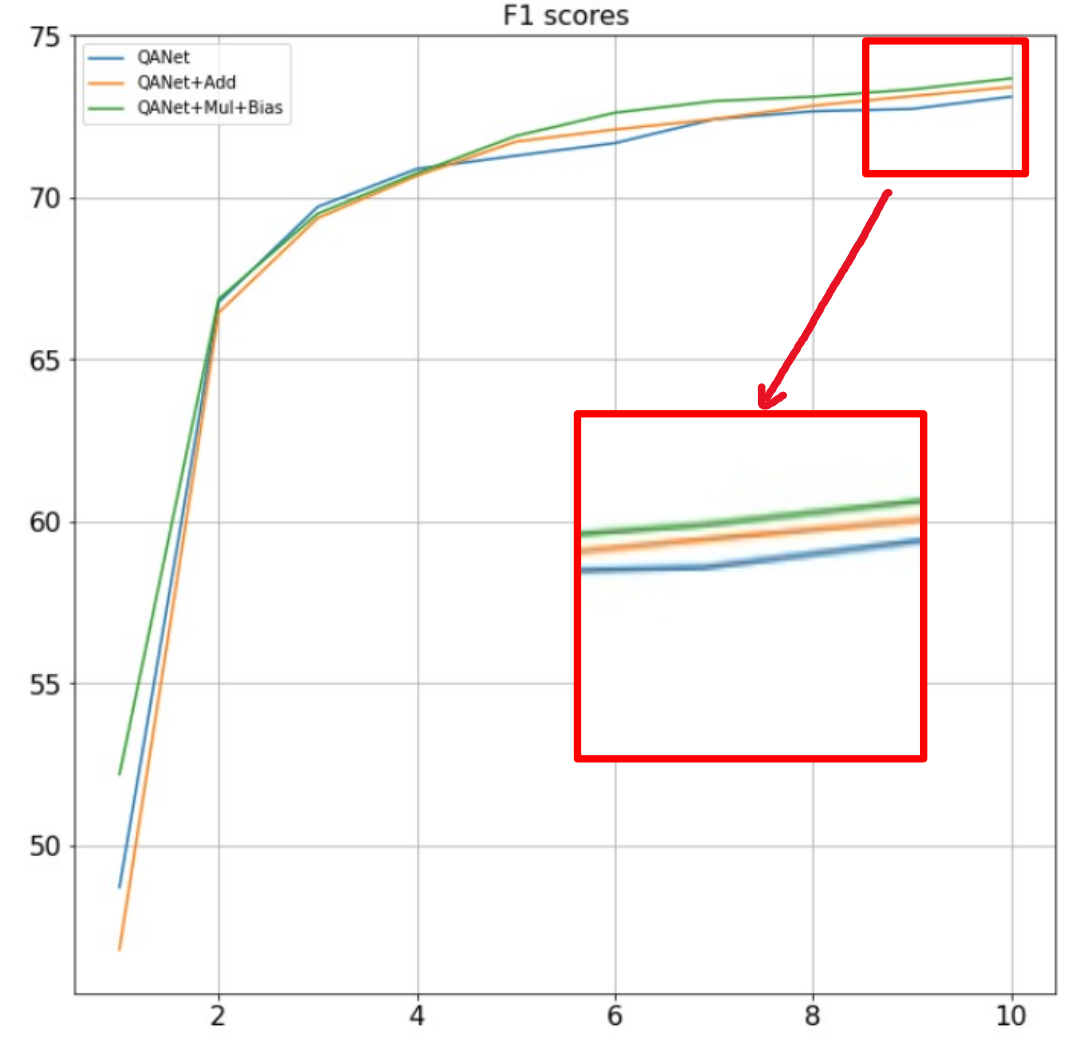
$$Z1 = ((W_2[M0; M2]); [X1 * Y1])$$

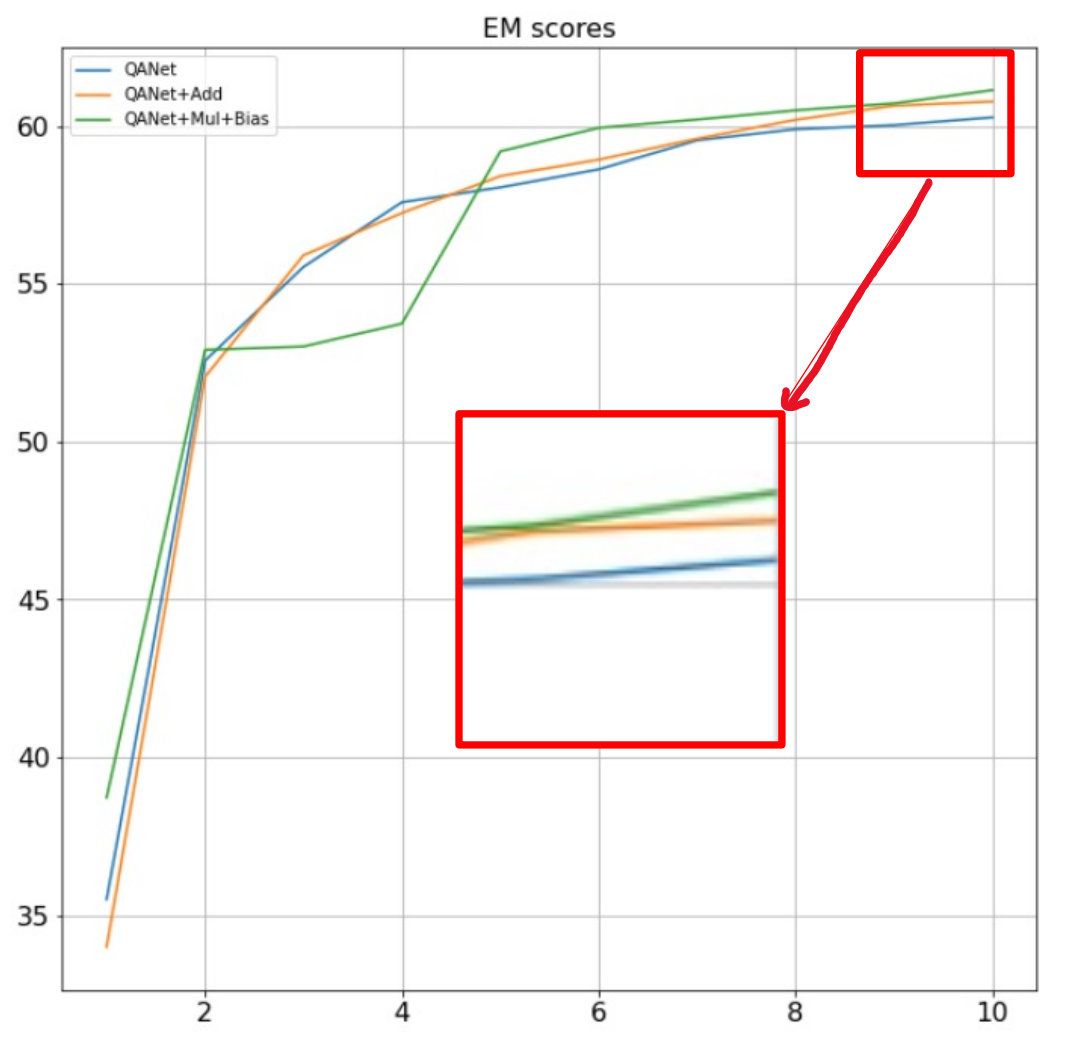
$$p^2 = Softmax\ (W_3 Z1 + b)$$



Result

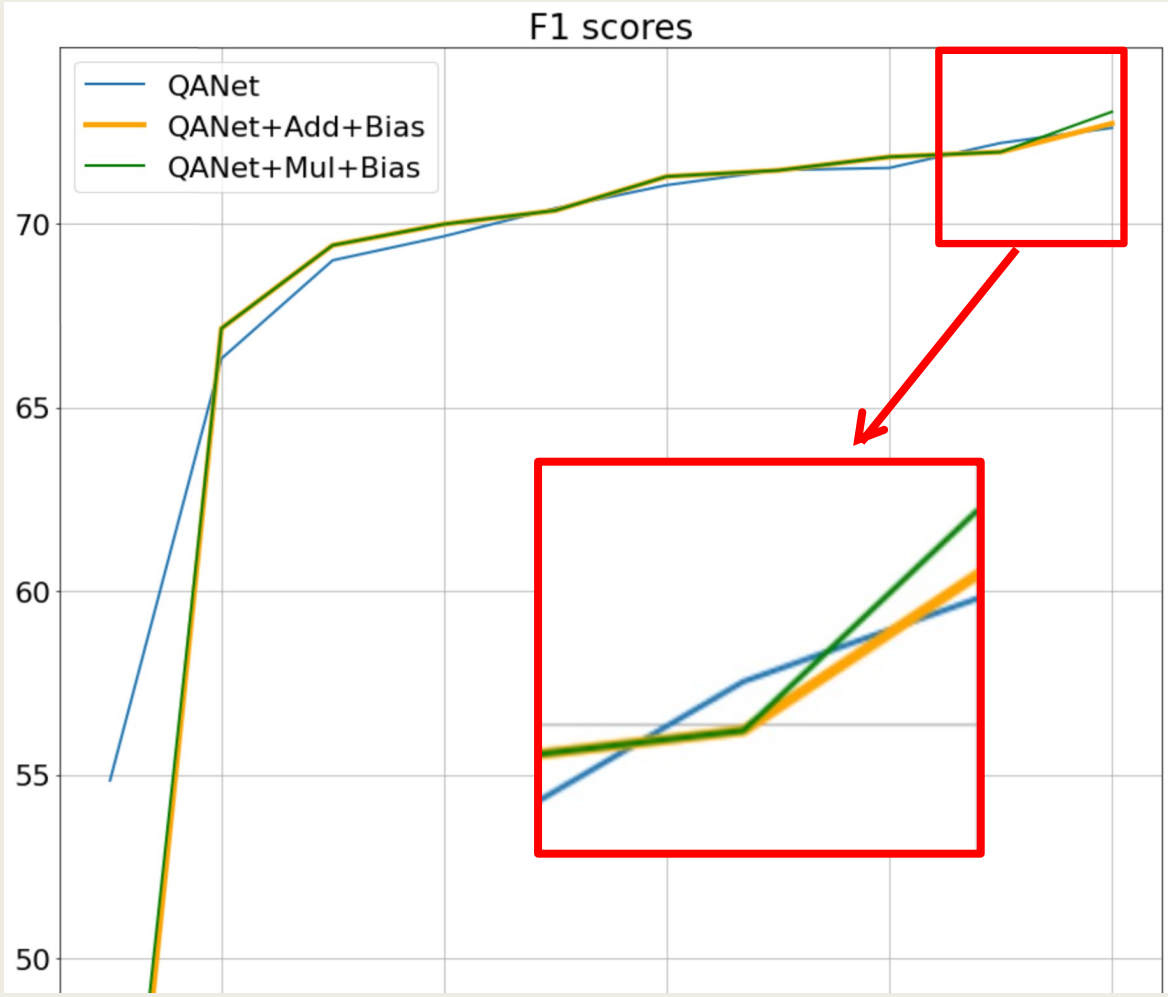
QANet vs. QANet+Add(1) vs. QANet+Multiplication+Bias(3)

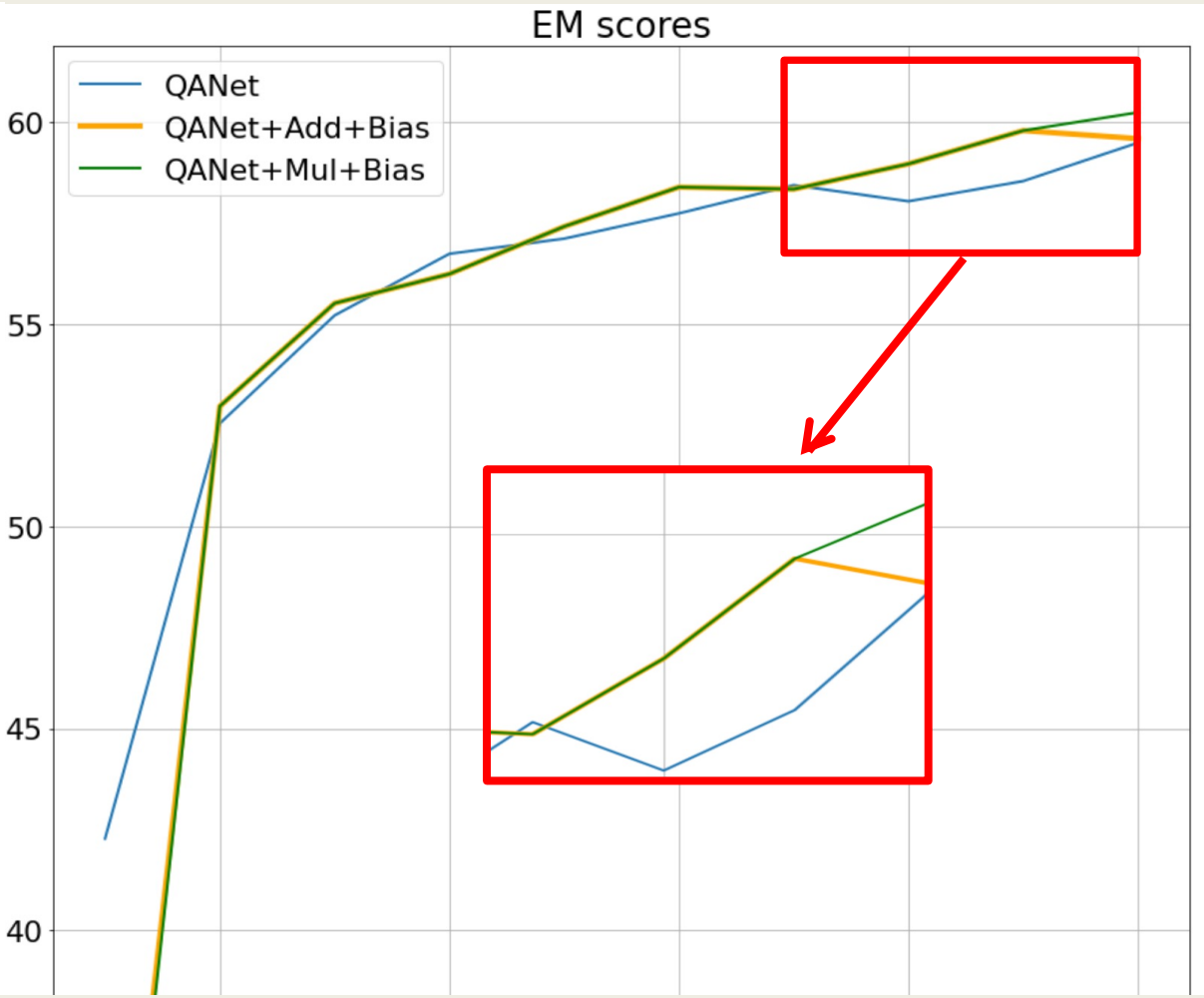




Result

QANet vs. QANet+Add+Bias(2) vs. QANet+Multiplication+Bias(3)





Conclusion & Analysis

Tests run on GPU **Tesla V100;** Batch size = **32**

	QANet	QANet+Add	QANet+Mul+Bias
F1 (# Epoch = 5)	71.29	71.72	71.90
EM (# Epoch = 5)	58.05	58.41	59.20
F1 (# Epoch = 10)	73.11	73.41	73.67
EM (# Epoch = 10)	60.27	60.78	61.14

Tests run on GPU **GeForce RTX 2080;** Batch size = **8**

	QANet	QANet+Add+Bias	QANet+Mul+Bias
F1 (# Epoch = 5)	70.43	71.84	70.37
EM (# Epoch = 5)	57.12	58.45	57.43
F1 (# Epoch = 10)	72.62	72.74	73.06
EM (# Epoch = 10)	59.50	59.60	60.25

- Reported EM/F1 = **73.6/82.7** in the QANet paper
- Did not converge yet but our model shows a tendency to have higher F1/EM scores
- More parameters to train, more epochs needed
- Future improvement: data augmentation using paraphrase

Reference

[1] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word//w representation. In Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[2] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[3] A. W. Yu *et al.*, “QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension.,” *CoRR*, vol. abs/1804.09541, 2018.

[4] Bi-Directional Attention Flow for Machine Comprehension (Minjoon Seo *et. al*, 2017)

[5] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, 2018.

[6] Hansika Hewamalage, Christoph Bergmeir, Kasun Bandara, Recurrent Neural Networks for Time Series Forecasting: Current status and future directions, International Journal of Forecasting, 2020

[7] Stanford NLP Group, The Stanford Question Answering Dataset.
URL: https://rajpurkar.github.io/SQuAD-explorer/

Q&A

Thanks for listening!