SQuAD

brettkoonce.com/talks

may 7th, 2019

overview

- background, squad, seq2seq
- bidaf, drqa, fastfusion/sru, qanet
- squad v2, transformer, ulmfit
- bert w/ squad v1, next steps

squad (v1)

- stanford question answer dataset
- question + context —> answer span
- 100k questions, human F1: 91.2%

Dataset	Example	Article / Paragraph
SQuAD	**Q**: How many provinces did the Ottoman empire contain in the 17th century? **A**: 32	**Article**: Ottoman Empire **Paragraph**: ... At the beginning of the 17th century the empire contained 32 provinces and numerous vassal states. Some of these were later absorbed into the Ottoman Empire, while others were granted various types of autonomy during the course of centuries.

seq2seq

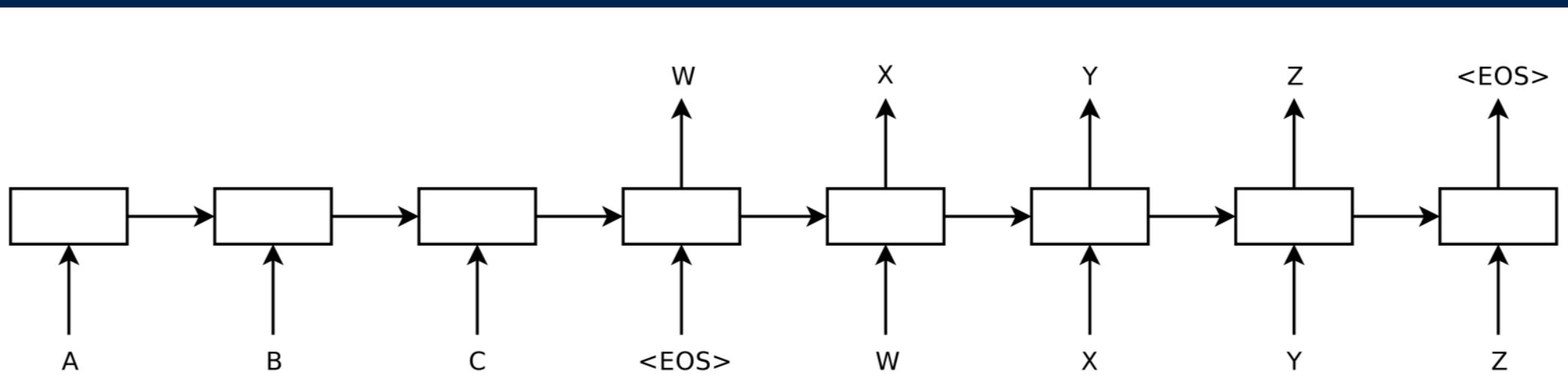


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

bidaf

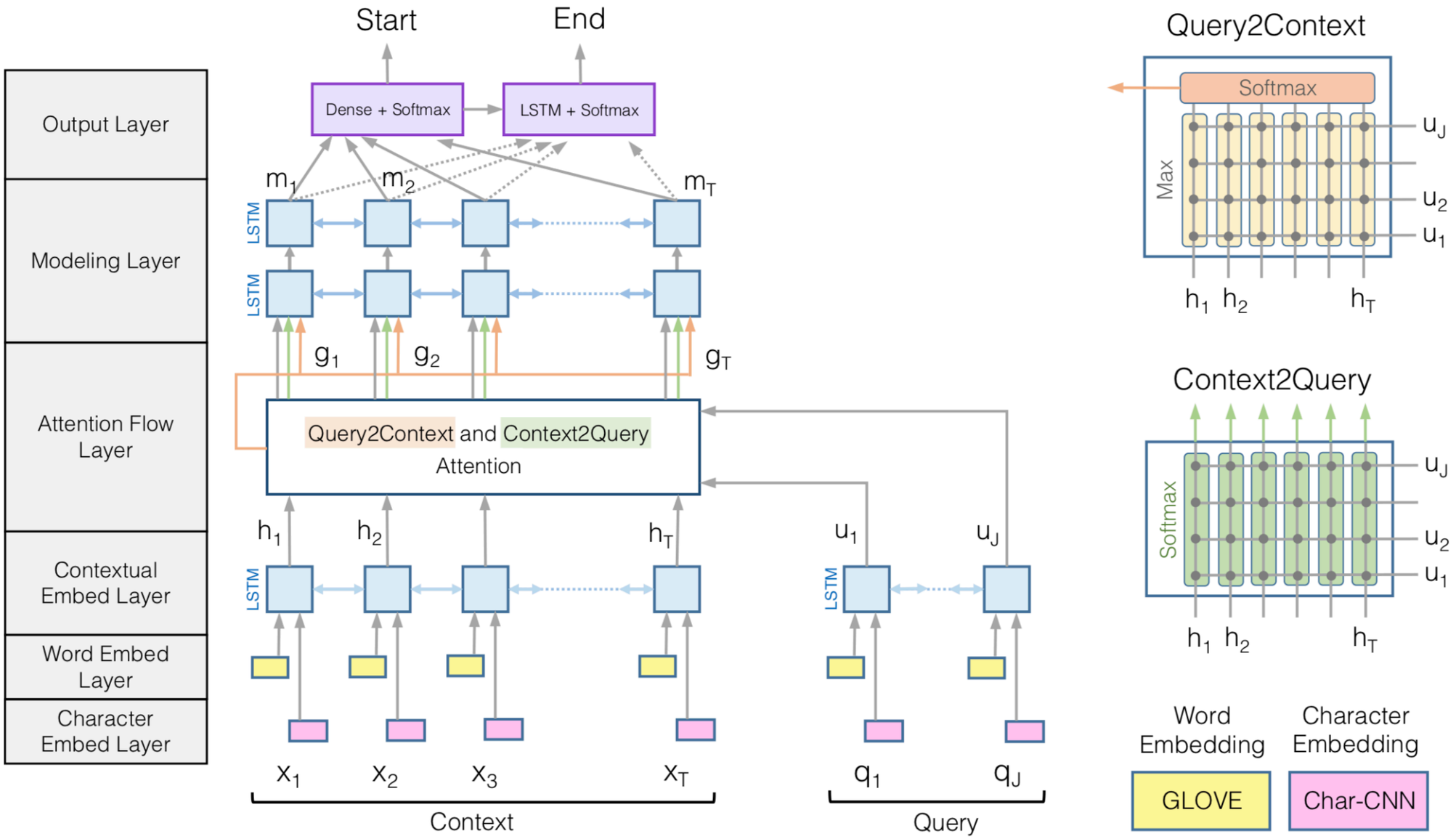


Figure 1: BiDirectional Attention Flow Model *(best viewed in color)*

drqa

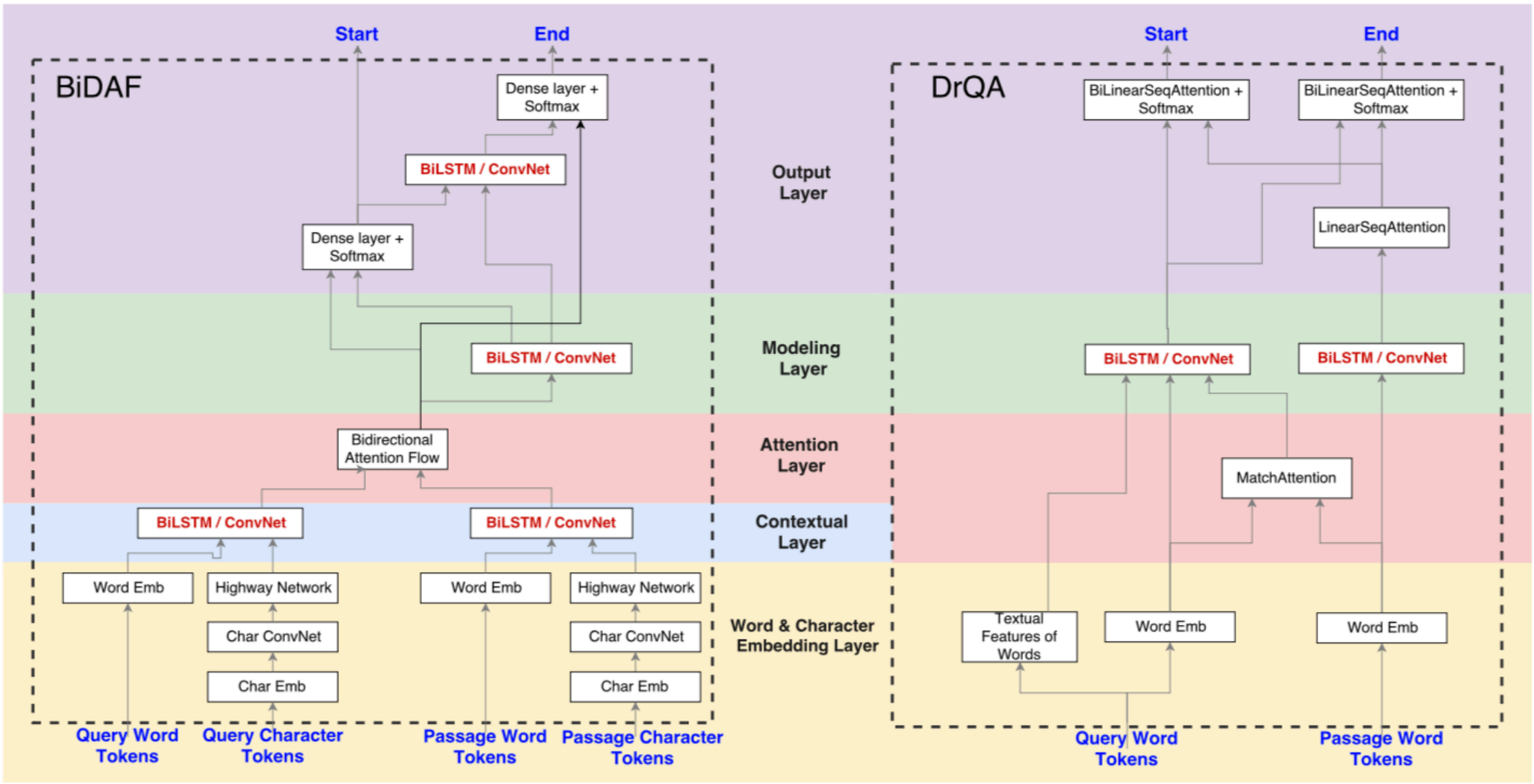


Figure 5: Schematic layouts of the BiDAF (*left*) and DrQA (*right*) architectures. We propose to replace all occurrences of BiLSTMs with diluted ConvNet structures.

fusionnet

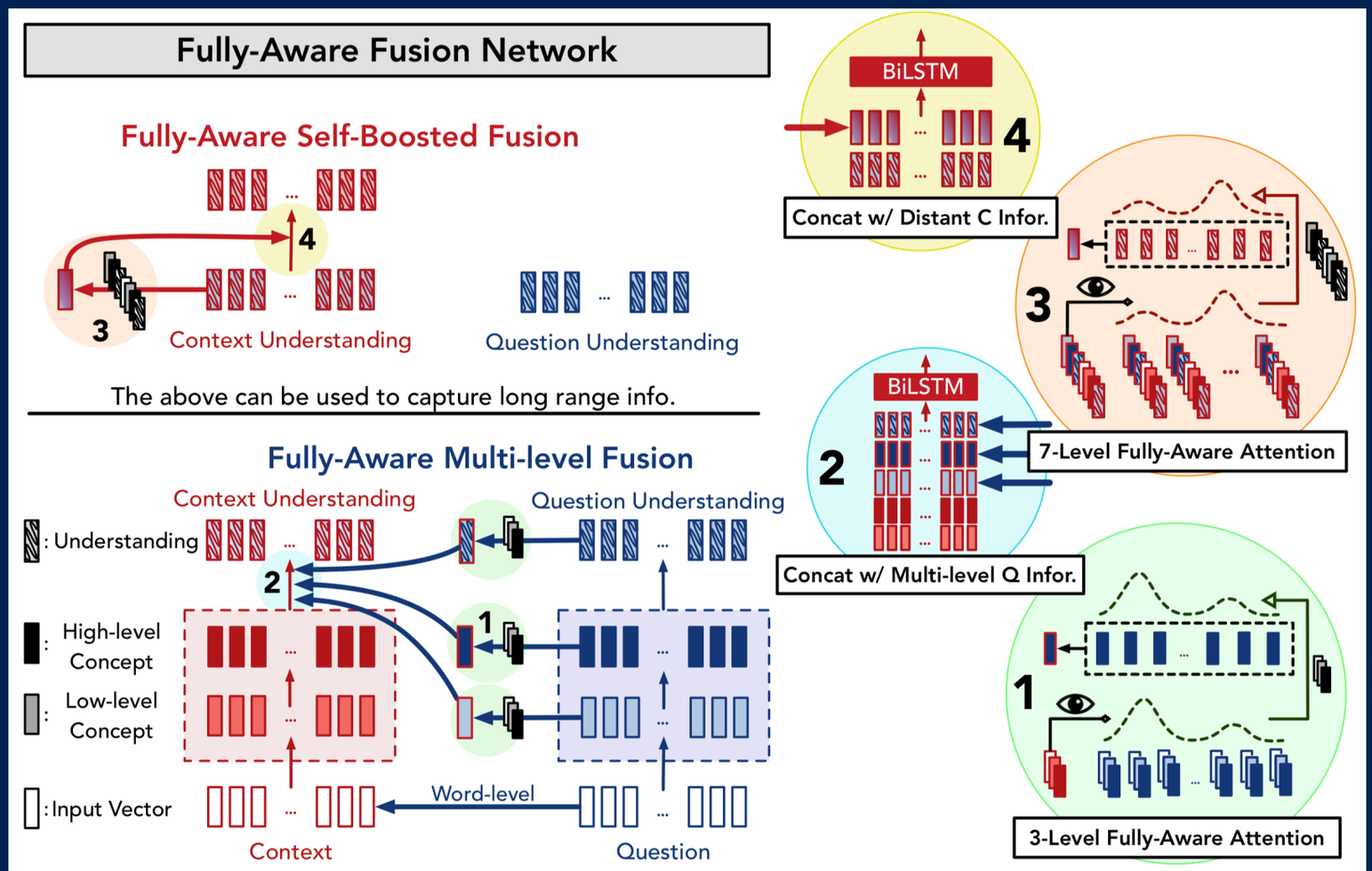


Figure 4: An illustration of FusionNet architecture. Each upward arrow represents one layer of BiLSTM. Each circle to the right is a detailed illustration of the corresponding component in FusionNet.
Circle 1: Fully-aware attention between C and Q. Illustration of Equation (C1) in Section 3.1.
Circle 2: Concatenate all concepts in C with multi-level Q information, then pass through BiLSTM. Illustration of Equation (C2) in Section 3.1.
Circle 3: Fully-aware attention on the context C itself. Illustration of Equation (C3) in Section 3.1.
Circle 4: Concatenate the understanding vector of C with self-attention information, then pass through BiLSTM. Illustration of Equation (C4) in Section 3.1.

simple recurrent unit

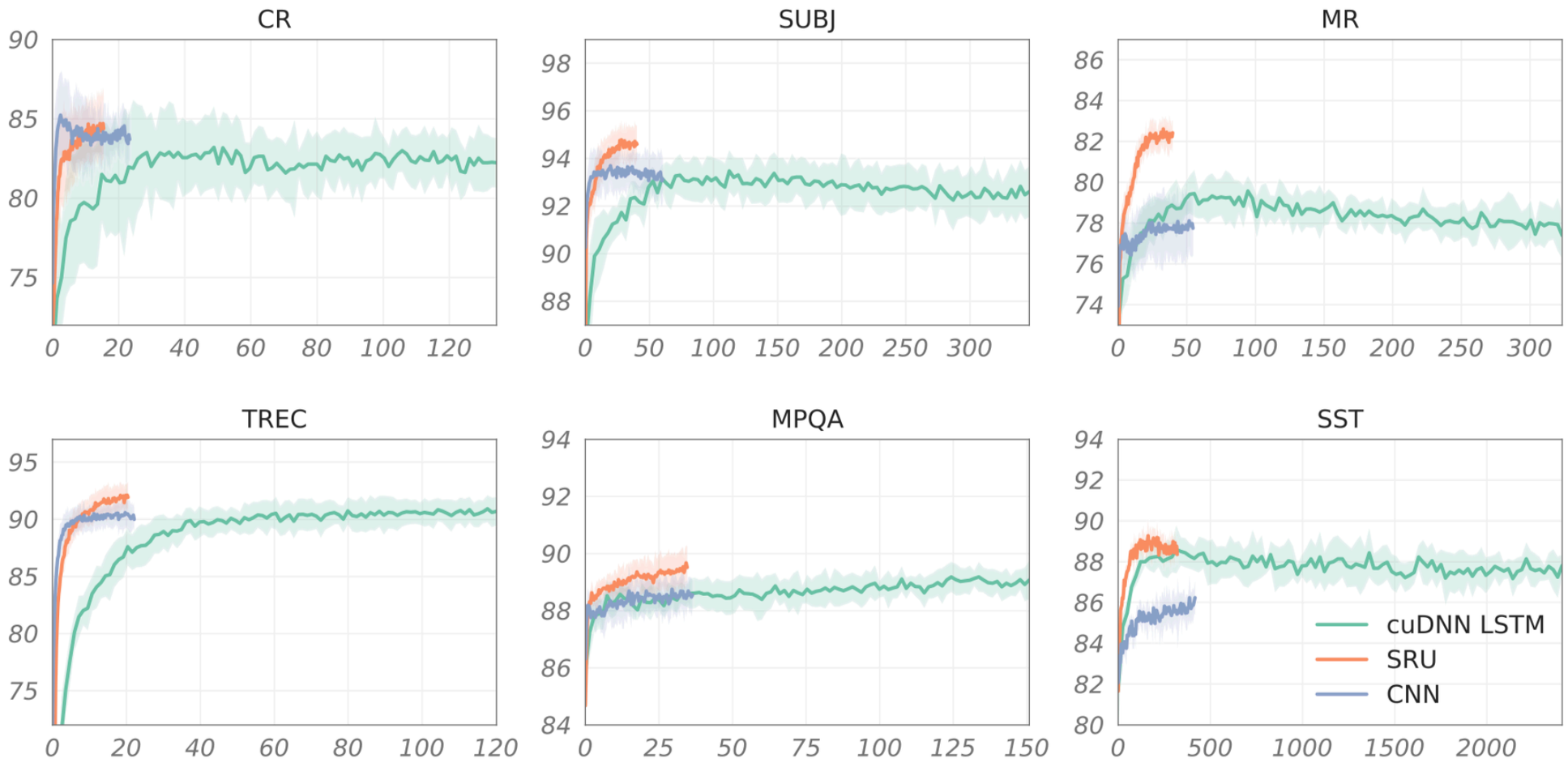


Figure 3: Mean validation accuracies (y-axis) and standard deviations of the CNN, 2-layer LSTM and 2-layer SRU models. We plot the curves of the first 100 epochs. X-axis is the training time used (in seconds). Timings are performed on NVIDIA GeForce GTX 1070 GPU, Intel Core i7-7700K Processor and cuDNN 7003.

qanet

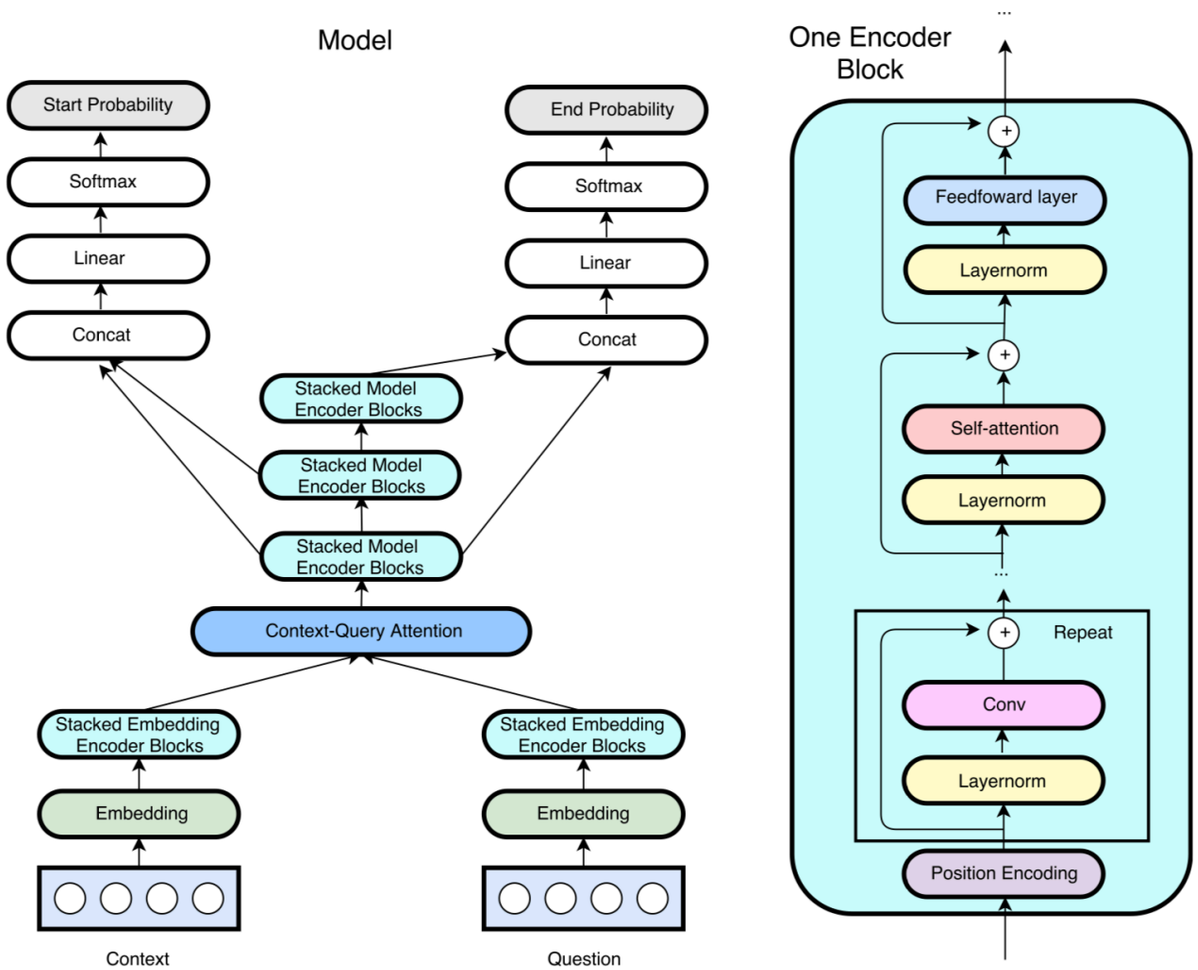


Figure 1: An overview of the QANet architecture (left) which has several Encoder Blocks. We use the same Encoder Block (right) throughout the model, only varying the number of convolutional layers for each block. We use layernorm and residual connection between every layer in the Encoder Block. We also share weights of the context and question encoder, and of the three output encoders. A positional encoding is added to the input at the beginning of each encoder layer consisting of sin and cos functions at varying wavelengths, as defined in (Vaswani et al., 2017a). 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.

dawnbench leaderboard

- **maximum:**
- **bidaf: 76%**
- **DrQA: 78%**
- **fusionnet + sru: 82%**
- **qanet: 84%**

Submission Date	Model	Time to 0.75 F1	Cost (USD)	Max F1 Score	Hardware	Framework
Mar 2019	FastFusionNet *Wu et al. (Cornell, SayMosaic, Google)* source	0:18:46	N/A	0.8236	1 NVidia GTX-1080 Ti	Pytorch v0.3.1
Dec 2018	DrQA *Runqi Yang, Facebook ParlAI, Brett Koonce* source	0:27:07	N/A	0.7829	1 NVidia 2080 RTX (dev box)	Pytorch 1.0.0
Apr 2018	QANet *Google* source	0:45:56	N/A	0.7637	1 TPUv2	TensorFlow v1.8
Dec 2018	DrQA *Runqi Yang, Facebook ParlAI, Brett Koonce* source	0:50:21	$0.76	0.7606	1 T4 / GCP	Pytorch 1.0.0
Dec 2018	DrQA *Runqi Yang, Facebook ParlAI, Brett Koonce* source	0:56:43	$0.57	0.7620	1 P4 / GCP	Pytorch 1.0.0
Sep 2018	DrQA *Runqi Yang, Facebook ParlAI, Brett Koonce* source	1:00:35	$3.09	0.7569	1 V100 / AWS p3.2xlarge	Pytorch 0.4.1
Sep 2018	DrQA *Runqi Yang, Facebook ParlAI, Brett Koonce* source	1:21:55	$1.23	0.7584	1 K80 / AWS p2.xlarge	Pytorch 0.4.1
Oct 2017	BiDAF *Stanford DAWN* source	7:38:10	$6.87	0.7533	1 K80 / 61 GB / 4 CPU (Amazon EC2 [p2.xlarge])	TensorFlow v1.2

squad v2

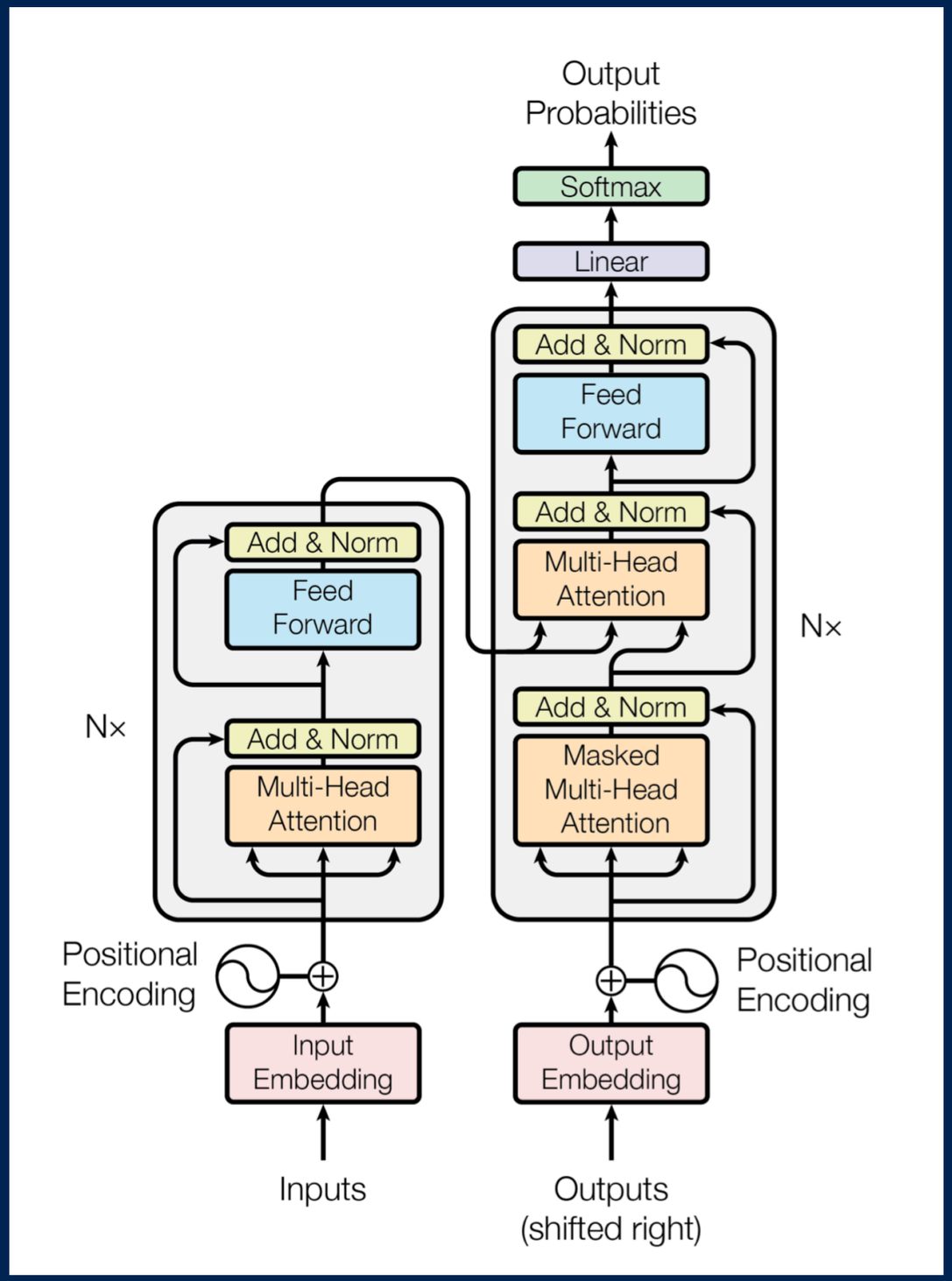
- new version of dataset
- adversarial questions (+50k)
- leaderboard

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph. How will your system compare to humans on this task?

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Mar 20, 2019	BERT + DAE + AoA (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	**87.147**	**89.474**
2 Mar 15, 2019	BERT + ConvLSTM + MTL + Verifier (ensemble) *Layer 6 AI*	86.730	89.286
3 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (ensemble) *Google AI Language* https://github.com/google-research/bert	86.673	89.147
4 Apr 13, 2019	SemBERT(ensemble) *Shanghai Jiao Tong University*	86.166	88.886
5 Mar 16, 2019	BERT + DAE + AoA (single model) *Joint Laboratory of HIT and iFLYTEK Research*	85.884	88.621
6 Mar 05, 2019	BERT + N-Gram Masking + Synthetic Self-Training (single model) *Google AI Language* https://github.com/google-research/bert	85.150	87.715
7 Jan 15, 2019	BERT + MMFT + ADA (ensemble) *Microsoft Research Asia*	85.082	87.615
7 Mar 13, 2019	BERT + ConvLSTM + MTL + Verifier (single model) *Layer 6 AI*	84.924	88.204

transformer



pretraining/transfer

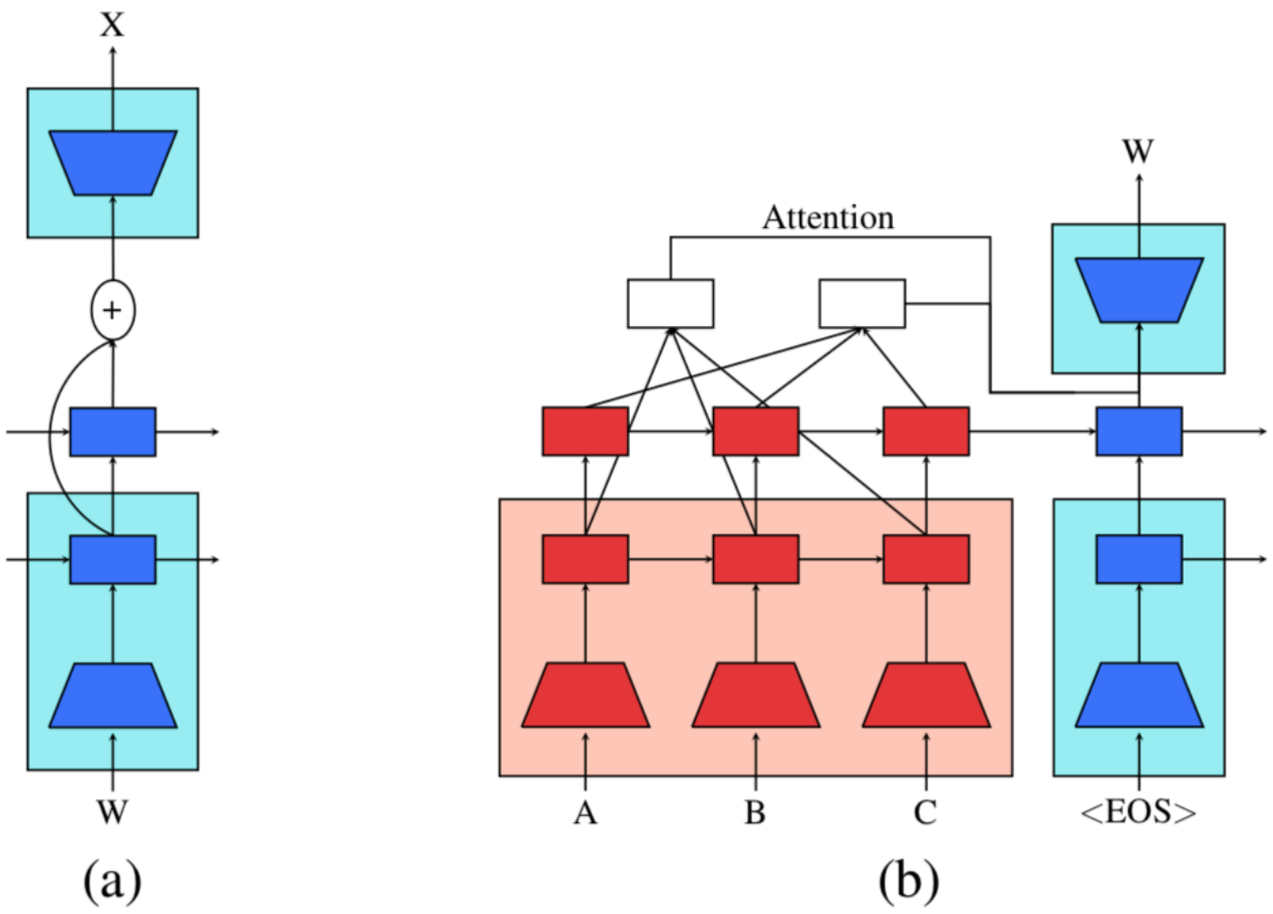


Figure 2: Two small improvements to the baseline model: (a) residual connection, and (b) multi-layer attention.

ulmfit

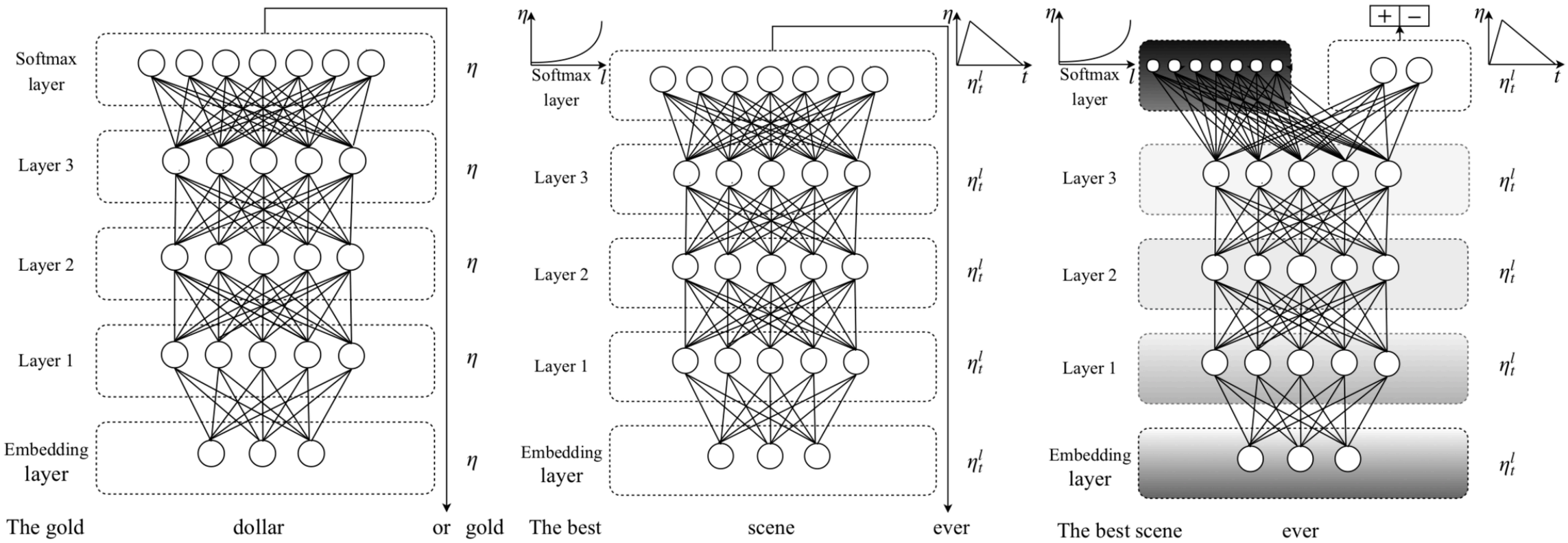


Figure 1: ULMFiT consists of three stages: a) The LM is trained on a general-domain corpus to capture general features of the language in different layers. b) The full LM is fine-tuned on target task data using discriminative fine-tuning ('*Discr*') and slanted triangular learning rates (STLR) to learn task-specific features. c) The classifier is fine-tuned on the target task using gradual unfreezing, '*Discr*', and STLR to preserve low-level representations and adapt high-level ones (shaded: unfreezing stages; black: frozen).

bert

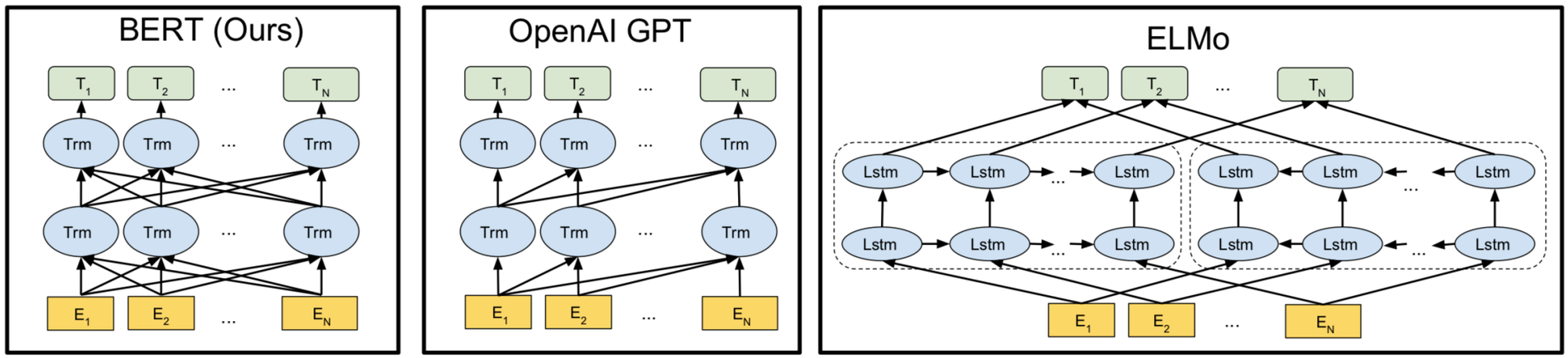


Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers.

beyond dawnbench

- **accuracy: retrained bert, 88% F1 (~4 hr, t4, huggingface)**
- **speed: qanet (tpu-2, <10 min)**
- **simplicity: runqi yang, hitvoice/drqa**
- **future: transformer-xl, gpt/gpt-2, mlperf**

thanks for coming!