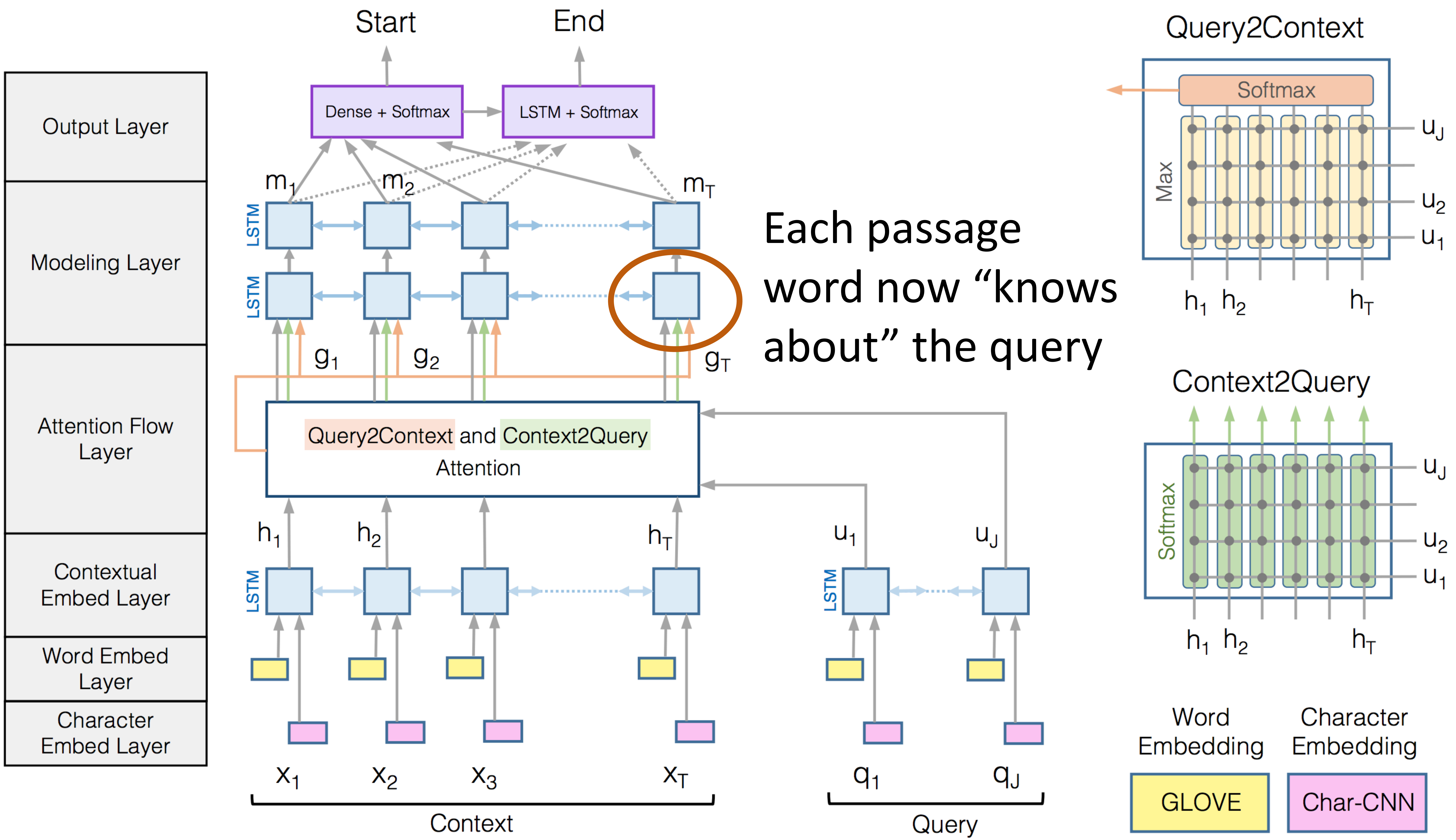
Reading Comprehension

Bidirectional Attention Flow



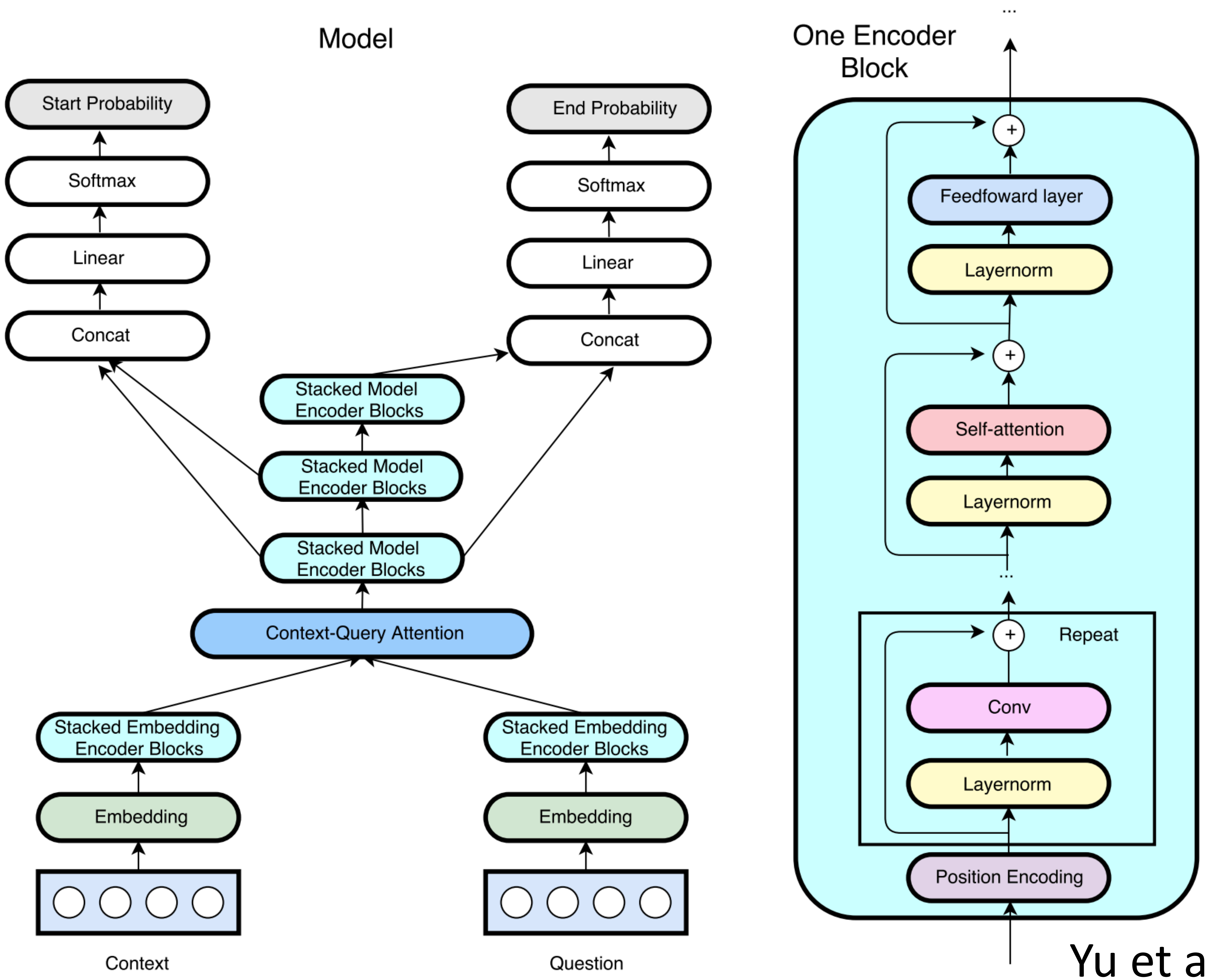
Each passage word now "knows about" the query

Seo et al. (2016)

QANet

- One of many models building on BiDAF in more complex ways

- Similar structure as BiDAF, but transformer layers (next lecture) instead of LSTMs



Yu et al. (2018)

SQuAD SOTA: Fall 18

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar et al. '16)	82.304	91.221
1 Oct 05, 2018	BERT (ensemble) *Google AI Language* https://arxiv.org/abs/1810.04805	**87.433**	**93.160**
2 Oct 05, 2018	BERT (single model) *Google AI Language* https://arxiv.org/abs/1810.04805	85.083	91.835
2 Sep 09, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.356	91.202
2 Sep 26, 2018	nlnet (ensemble) *Microsoft Research Asia*	85.954	91.677
3 Jul 11, 2018	QANet (ensemble) *Google Brain & CMU*	84.454	90.490
4 Jul 08, 2018	r-net (ensemble) *Microsoft Research Asia*	84.003	90.147
5 Mar 19, 2018	QANet (ensemble) *Google Brain & CMU*	83.877	89.737

- BiDAF: 73 EM / 81 F1
- nlnet, QANet, r-net — dueling super complex systems (much more than BiDAF…)

SQuAD 2.0 SOTA: Spring 2019

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

- Harder variant of SQuAD
- Since spring 2019: SQuAD performance is dominated by large pre-trained models like BERT

Adversarial Examples

- Can construct adversarial examples that fool these systems: add one carefully chosen sentence and performance drops to below 50%
- Still "surface-level" matching, not complex understanding
- Other challenges: recognizing when answers aren't present, doing multi-step reasoning

Article: Super Bowl 50
Paragraph: "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
Question: "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

Jia and Liang (2017)

Pre-training / ELMo

What is pre-training?

- "Pre-train" a model on a large dataset for task X, then "fine-tune" it on a dataset for task Y
- Key idea: X is somewhat related to Y, so a model that can do X will have some good neural representations for Y as well
- ImageNet pre-training is huge in computer vision: learn generic visual features for recognizing objects
- GloVe can be seen as pre-training: learn vectors with the skip-gram objective on large data (task X), then fine-tune them as part of a neural network for sentiment/any other task (task Y)

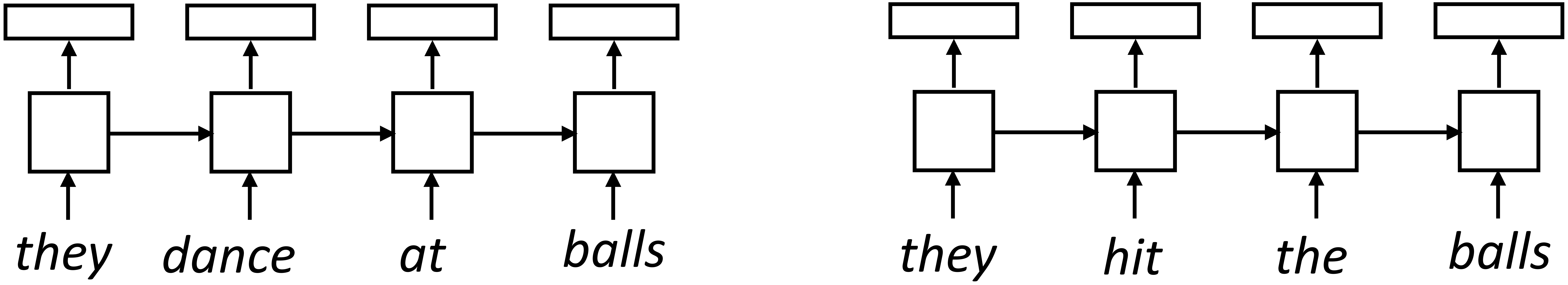
GloVe is insufficient

- GloVe uses a lot of data but in a weak way
  - Take a powerful language model, train it on large amounts of data, then use those representations in downstream tasks
- Having a single embedding for each word is wrong

they dance at balls   *they hit the balls*

  - Identifying discrete word senses is hard, doesn't scale. Hard to identify how many senses each word has
- How can we make our word embeddings more *context-dependent*?

Context-dependent Embeddings



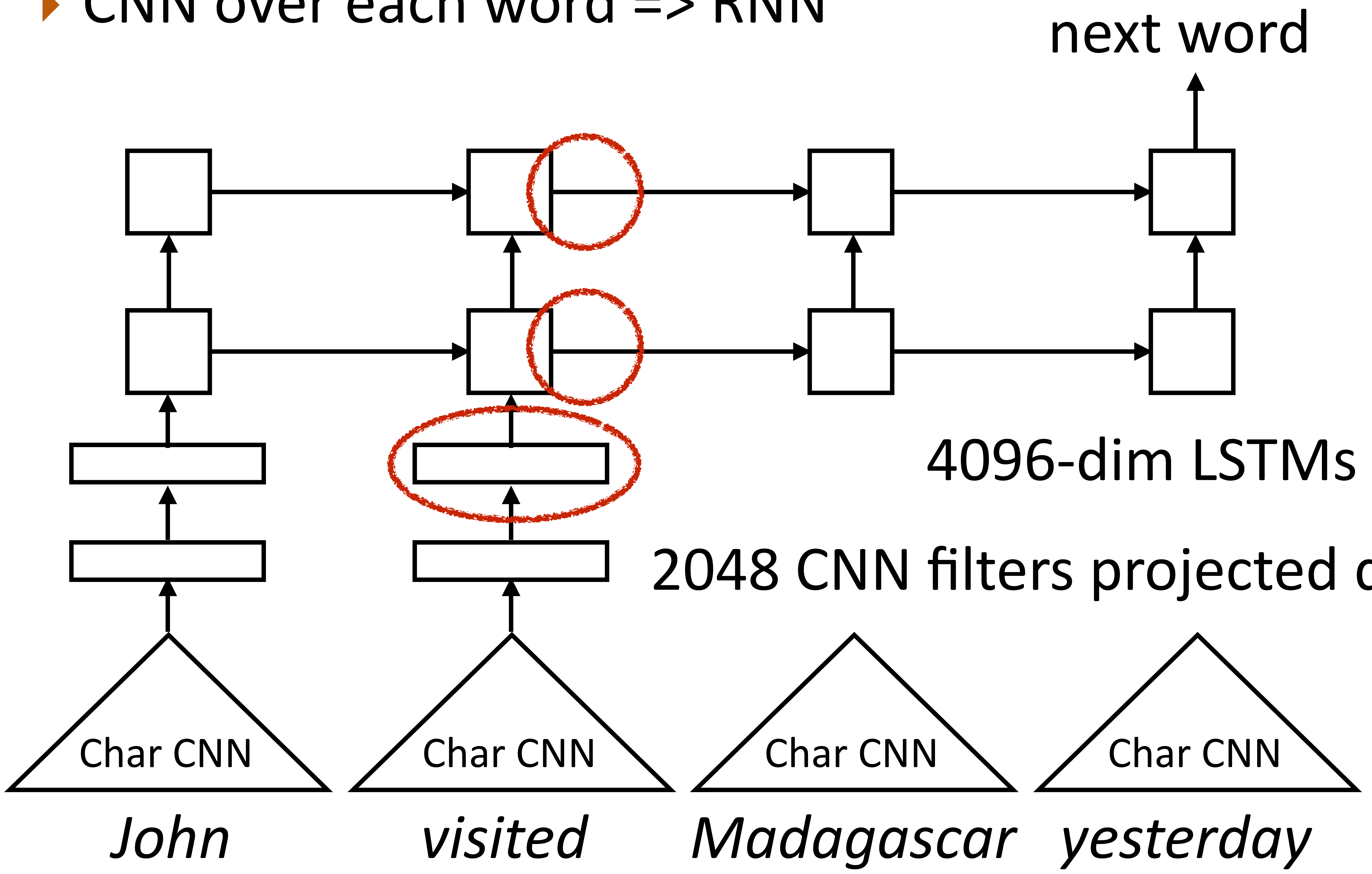
- Train a neural language model to predict the next word given previous words in the sentence, use the hidden states (output) at each step *as word embeddings*

- This is the key idea behind ELMo: language models can allow us to form useful word representations in the same way word2vec did

Peters et al. (2018)

ELMo

- CNN over each word => RNN



Representation of *visited* (plus vectors from another LM running backwards)

*getting this model right took **years**

Peters et al. (2018)

Training ELMo 

- Data: 1B Word Benchmark (Chelba et al., 2014)

- Pre-training time: 2 weeks on 3 NVIDIA GTX 1080 GPUs

  - Much lower time cost if we used V100s / Google's TPUs, but still hundreds of dollars in compute cost to train once

  - Larger BERT models trained on more data (next week) cost $10k+

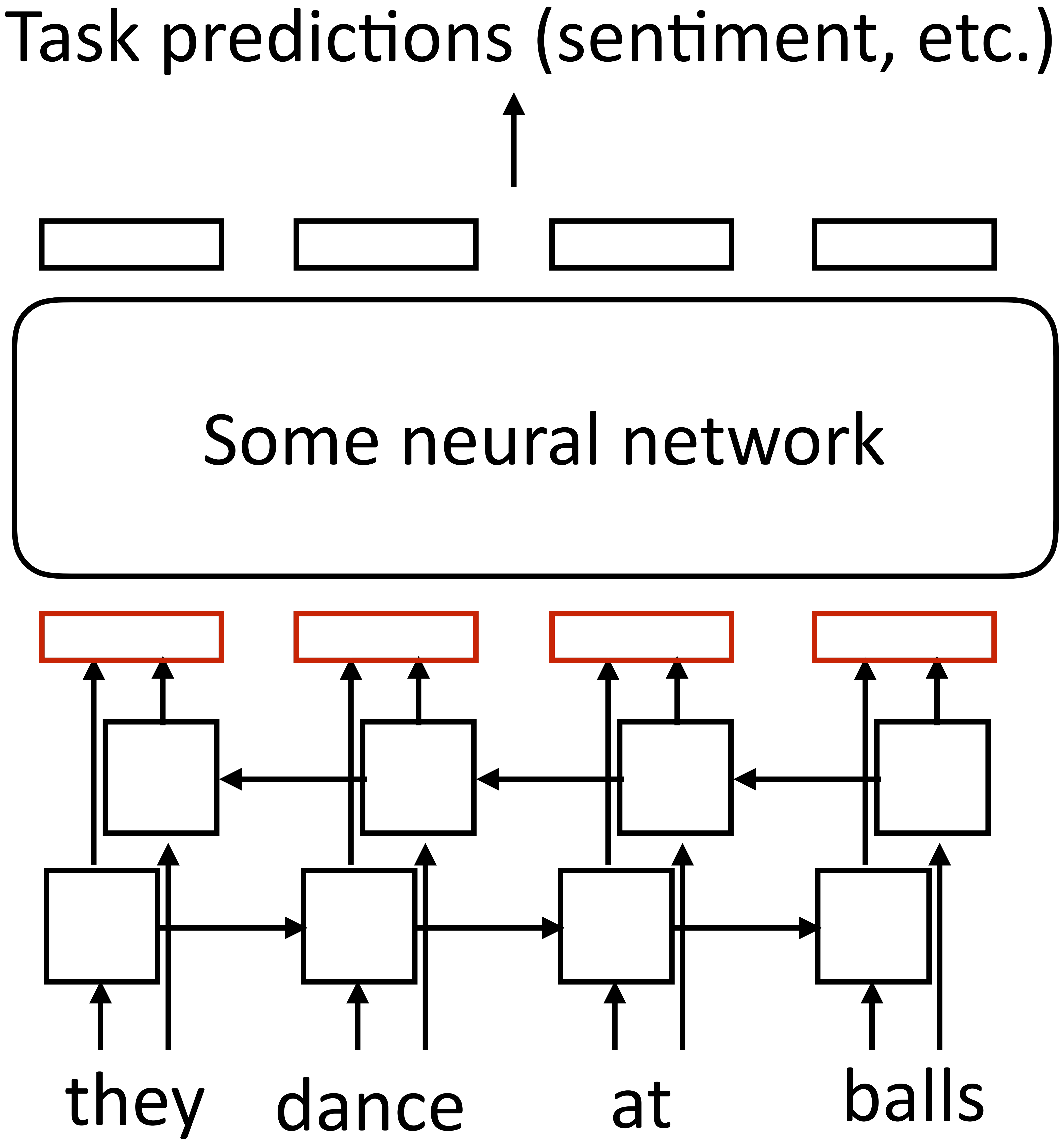
- Pre-training is expensive, but fine-tuning is doable

How to apply ELMo?

- Take those embeddings and feed them into whatever architecture you want to use for your task

- *Frozen* embeddings (most common): update the weights of your network but keep ELMo's parameters frozen

- *Fine-tuning*: backpropagate all the way into ELMo when training your model



Results: Frozen ELMo

TASK	PREVIOUS SOTA		OUR BASELINE	ELMO + BASELINE	INCREASE (ABSOLUTE/ RELATIVE)
SQuAD	Liu et al. (2017)	84.4	81.1	85.8	4.7 / 24.9%
SNLI	Chen et al. (2017)	88.6	88.0	88.7 ± 0.17	0.7 / 5.8%
SRL	He et al. (2017)	81.7	81.4	84.6	3.2 / 17.2%
Coref	Lee et al. (2017)	67.2	67.2	70.4	3.2 / 9.8%
NER	Peters et al. (2017)	91.93 ± 0.19	90.15	92.22 ± 0.10	2.06 / 21%
SST-5	McCann et al. (2017)	53.7	51.4	54.7 ± 0.5	3.3 / 6.8%

QA

(sort of) like dep parsing

Five-class version of sentiment from A1-A2

- Massive improvements, beating models handcrafted for each task
- These are mostly *text analysis* tasks. Other pre-training approaches needed for text generation like translation

Peters et al. (2018)

Why is language modeling a good objective?

- "Impossible" problem but bigger models seem to do better and better at distributional modeling (no upper limit yet)

- Successfully predicting next words requires modeling lots of different effects in text

Context: My wife refused to allow me to come to Hong Kong when the plague was at its height and –" "Your wife, Johanne? You are married at last ?" Johanne grinned. "Well, when a man gets to my age, he starts to need a few home comforts.

Target sentence: After my dear mother passed away ten years ago now, I became _____.

Target word: lonely

Probing ELMo

- From each layer of the ELMo model, attempt to predict something: POS tags, word senses, etc.
- Higher accuracy => ELMo is capturing that thing more strongly

Model	F_1
WordNet 1st Sense Baseline	65.9
Raganato et al. (2017a)	69.9
Iacobacci et al. (2016)	**70.1**
CoVe, First Layer	59.4
CoVe, Second Layer	*64.7*
biLM, First layer	67.4
biLM, Second layer	*69.0*

Table 5: All-words fine grained WSD F_1. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.

Model	Acc.
Collobert et al. (2011)	97.3
Ma and Hovy (2016)	97.6
Ling et al. (2015)	**97.8**
CoVe, First Layer	*93.3*
CoVe, Second Layer	92.8
biLM, First Layer	*97.3*
biLM, Second Layer	96.8

Table 6: Test set POS tagging accuracies for PTB. For CoVe and the biLM, we report scores for both the first and second layer biLSTMs.

Peters et al. (2018)

Analysis

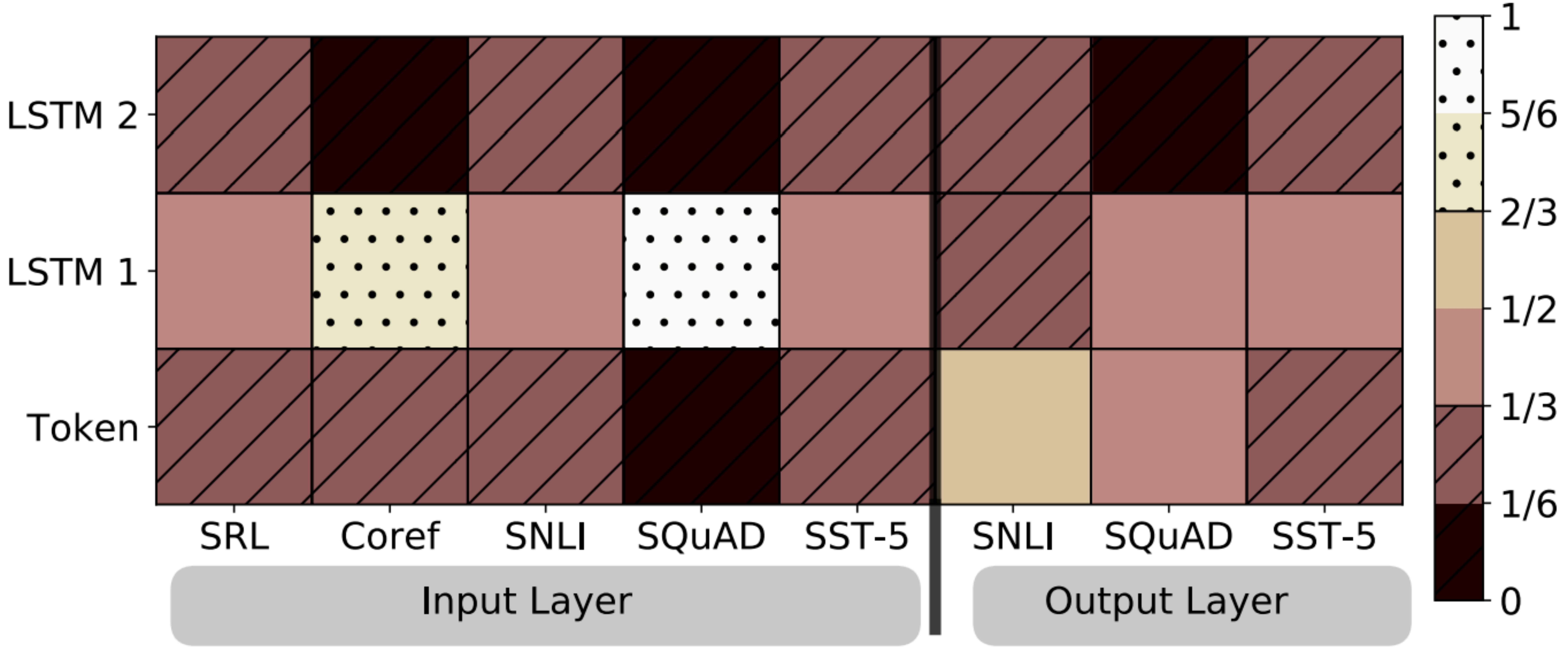


Figure 2: Visualization of softmax normalized biLM layer weights across tasks and ELMo locations. Normalized weights less then 1/3 are hatched with horizontal lines and those greater then 2/3 are speckled.

Peters et al. (2018)

Takeaways

- Learning a large language model can be an effective way of generating "word embeddings" informed by their context
- Pre-training on massive amounts of data can improve performance on tasks like QA
- Next class: transformers and BERT