

CS11-711 Advanced NLP

# Document Level Models

Daniel Fried



**Carnegie Mellon University**  
Language Technologies Institute

Site

<https://cmu-anlp.github.io/>

(w/ slides from Graham Neubig and Zhengzhong Liu)

# Some NLP Tasks we've Handled

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

$$P(w_{i+1} = \text{of} \mid w_i = \text{tired}) = 1$$

$$P(w_{i+1} = \text{of} \mid w_i = \text{use}) = 1$$

$$P(w_{i+1} = \text{sister} \mid w_i = \text{her}) = 1$$

$$P(w_{i+1} = \text{beginning} \mid w_i = \text{was}) = 1/2$$

$$P(w_{i+1} = \text{reading} \mid w_i = \text{was}) = 1/2$$

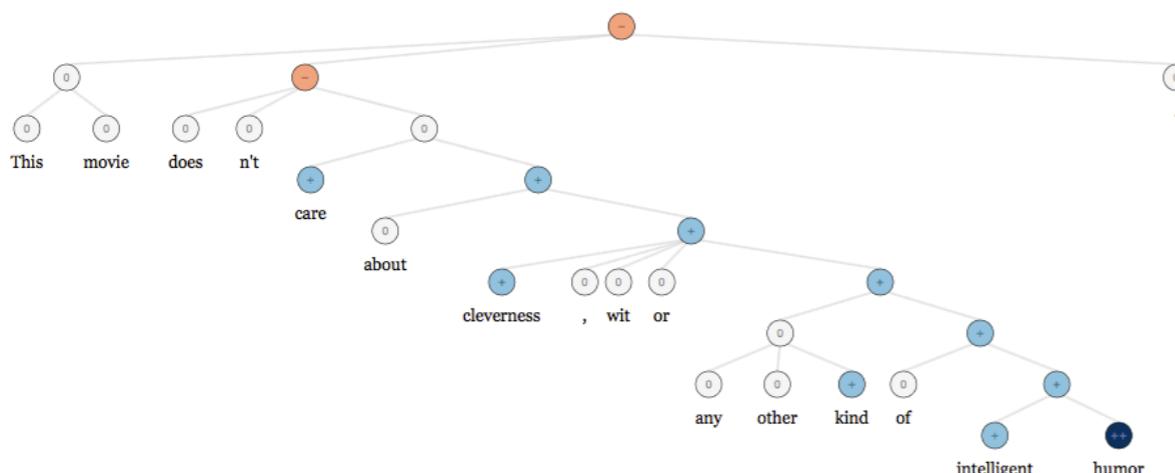
$$P(w_{i+1} = \text{bank} \mid w_i = \text{the}) = 1/3$$

$$P(w_{i+1} = \text{book} \mid w_i = \text{the}) = 1/3$$

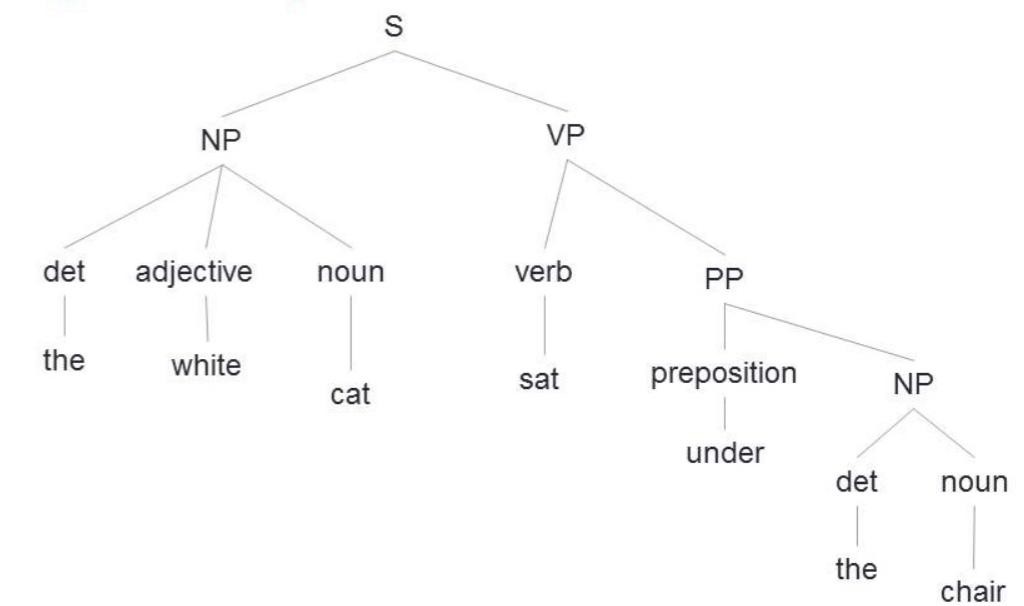
$$P(w_{i+1} = \text{use} \mid w_i = \text{the}) = 1/3$$

$$P(X|Y) = \frac{P(X,Y)}{P(Y)}$$

Language Models



Classification



Parsing

Germany's representative to the European Union's veterinary committee Werner Zwingman said on Wednesday consumers should ...

Entity Tagging

# Some Connections to Tasks over Documents

Prediction using documents

- **Document-level language modeling:** Predicting language on the multi-sentence level (c.f. single-sentence language modeling)
- **Document classification:** Predicting traits of entire documents (c.f. sentence classification)

- **Entity coreference:** Which entities correspond to each-other? (c.f. NER)
- **Discourse parsing:** How do segments of a document correspond to each-other? (c.f. syntactic parsing)

Prediction of document structure

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

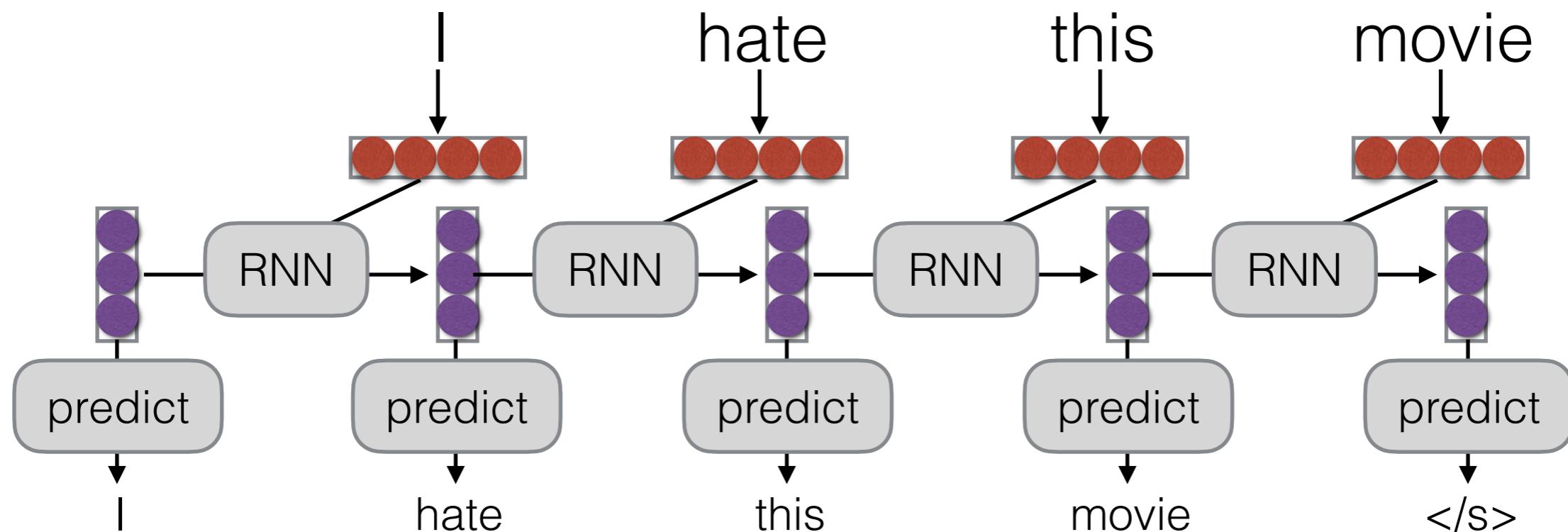
# Document Level Language Modeling

# Document Level Language Modeling

- We want to predict the probability of words in an entire document
- Obviously sentences in a document don't exist in a vacuum! We want to take advantage of this fact.

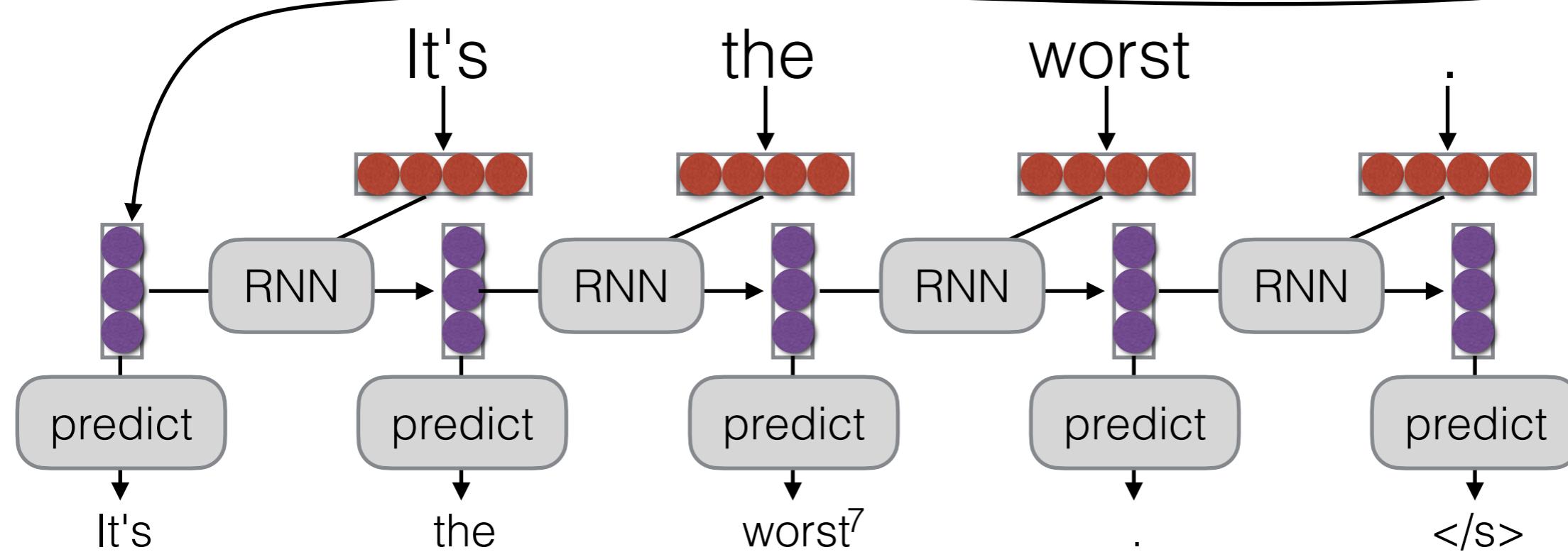
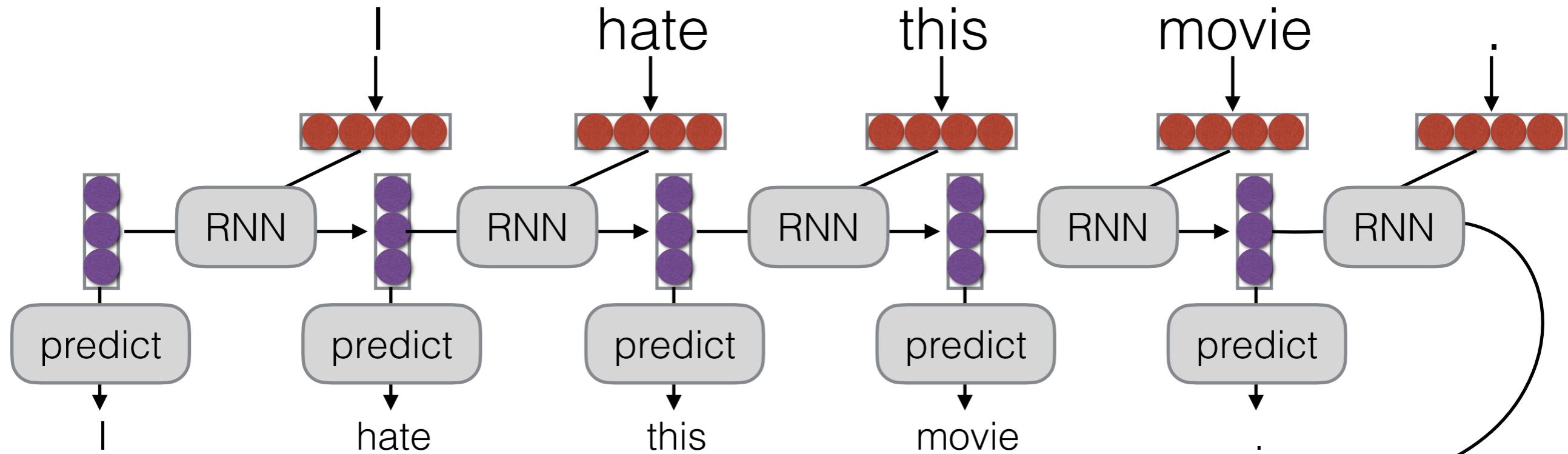
# Remember: Modeling using Recurrent Networks

- Model passing previous information in hidden state



# Simple: Infinitely Pass State

(Mikolov et al. 2011)



# Separate Encoding for Coarse-grained Document Context

(Mikolov & Zweig 2012)

- One big RNN for local and global context tends to miss out on global context (as local context is more predictive)
- Other attempts try to incorporate document-level context explicitly

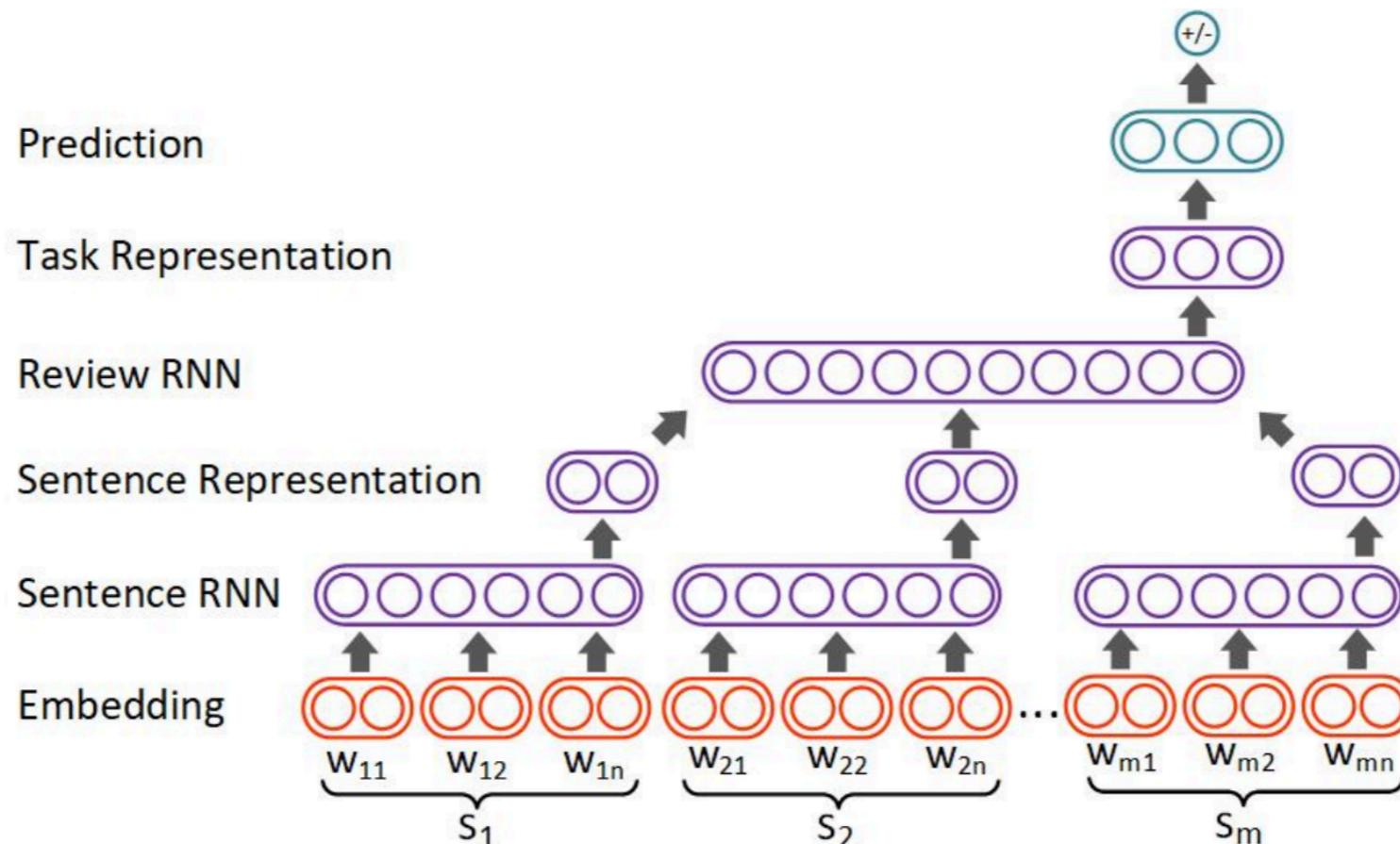
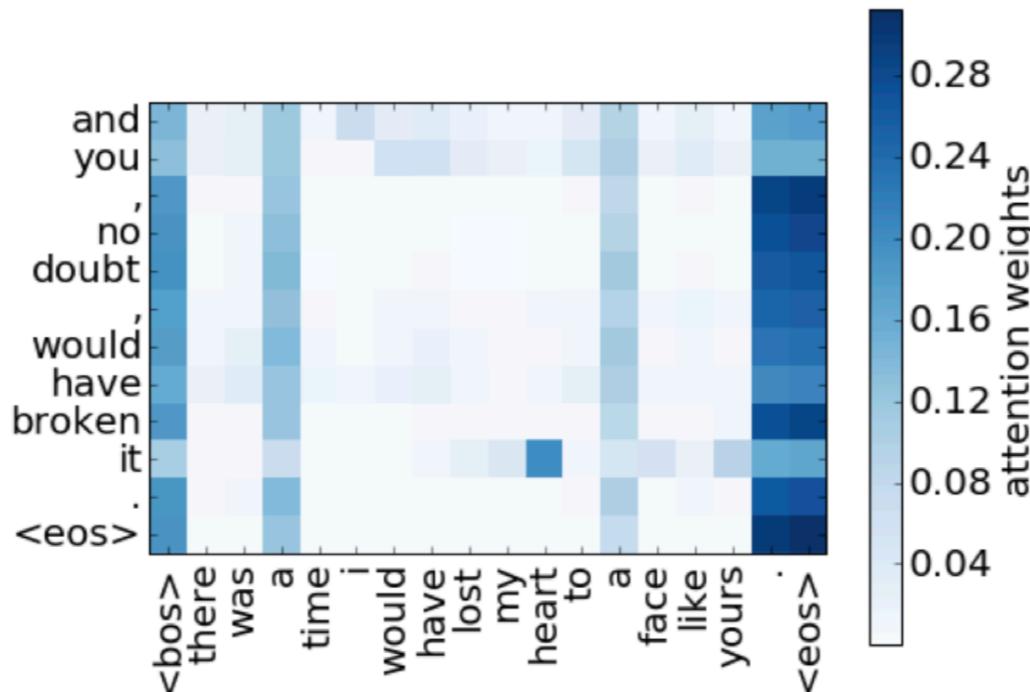


Figure from  
Jauhar et al. 2018

# Self-attention/Transformers Across Sentences

- Simply self-attend to all previous words in the document (e.g. Voita et al. 2018)
- + Can relatively simply use document-level context
- + Can learn interesting phenomena (e.g. co-reference)



- - Computation is quadratic in sequence length!

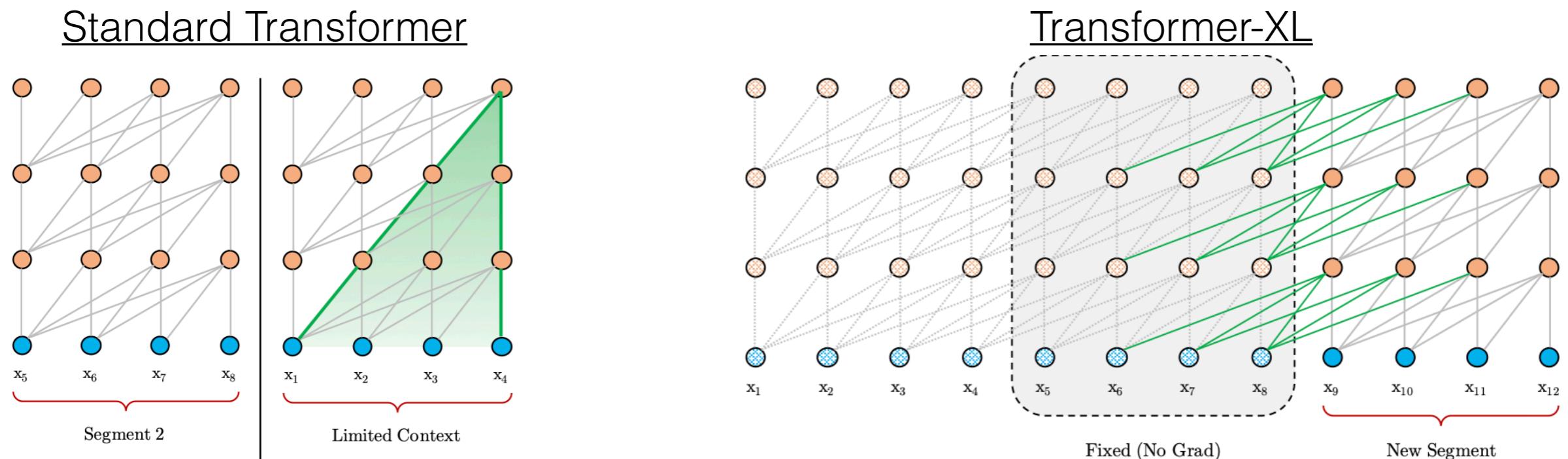
# Efficient Transformers

# Transformer-XL:

## Truncated BPTT+Transformer

(Dai et al. 2019)

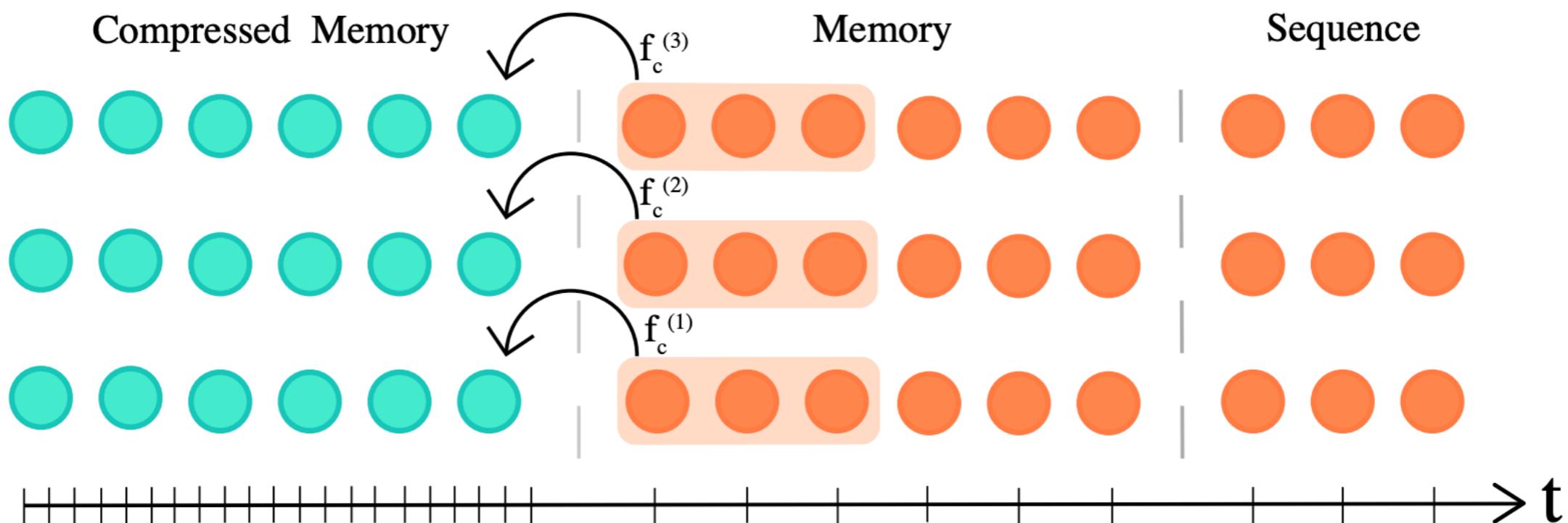
- Idea: attend to fixed **vectors** from the previous sentence (Dai et al. 2019)



- Like truncated backprop through time for RNNs; can use previous states, but not backprop into them

# Compressing Previous States

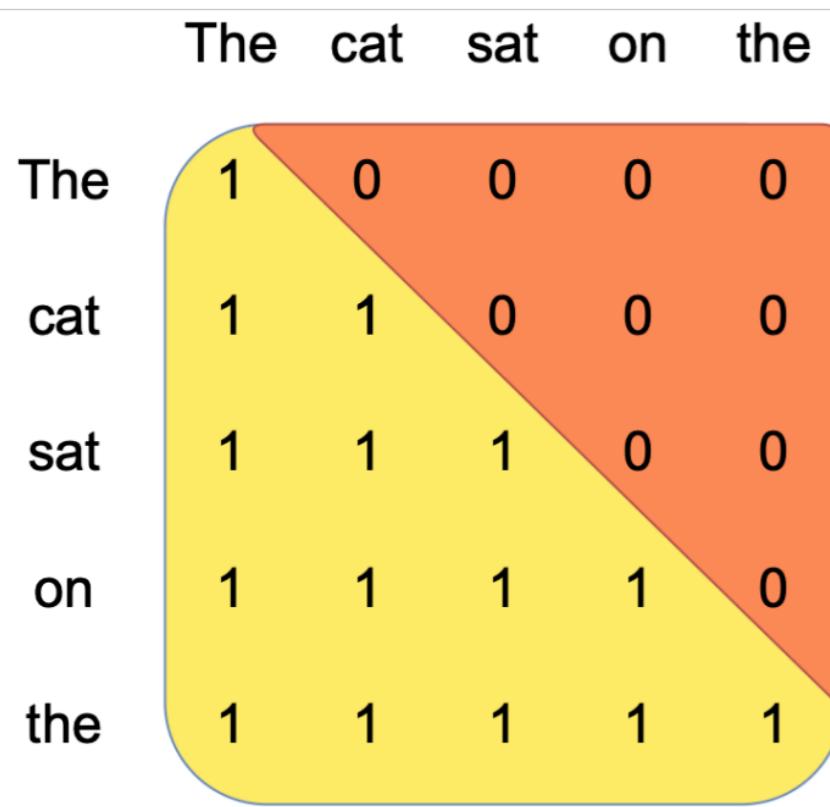
- Add a "strided" compression step over previous states (Lillicrap et al. 2019)



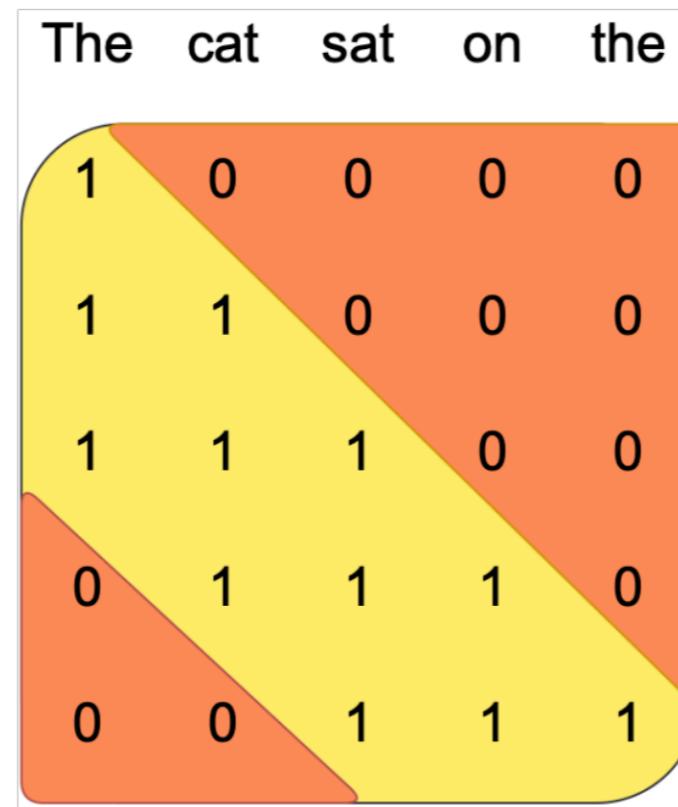
# Sparse Transformers

(Child et al. 2019, Beltagy et al. 2020)

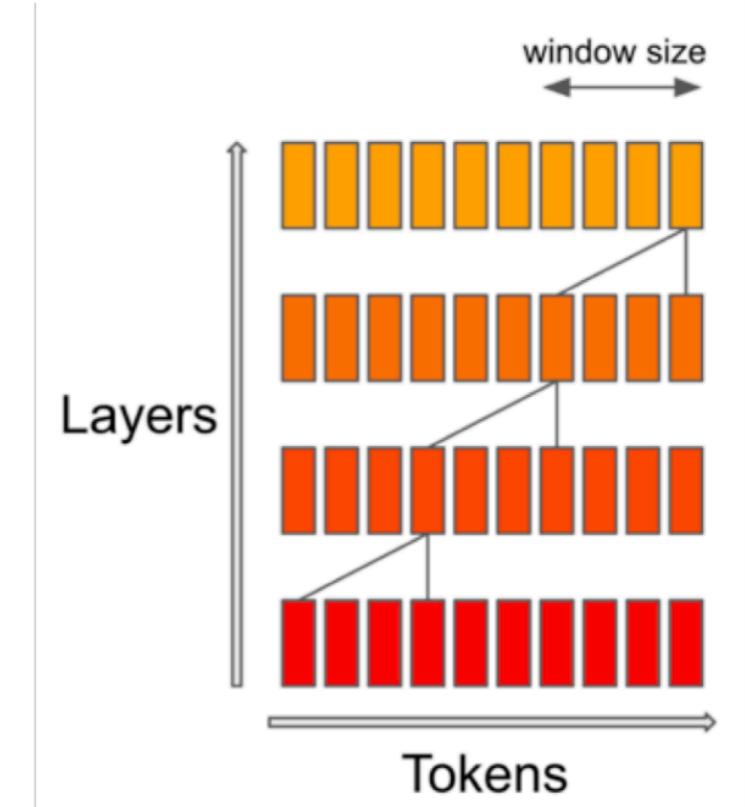
- Add “stride” / sliding window, only attending to every  $n$  previous states
- Requires special operations, often via custom CUDA kernels



**Vanilla Attention**



**Sliding Window Attention**



**Effective Context Length**

Figure from Mistral-7B paper, Jiang et al. 2023

# Adaptive Span Transformers

- Can make the span adaptive attention head by attention head some are short, some long (Sukhbaatar et al. 2019)

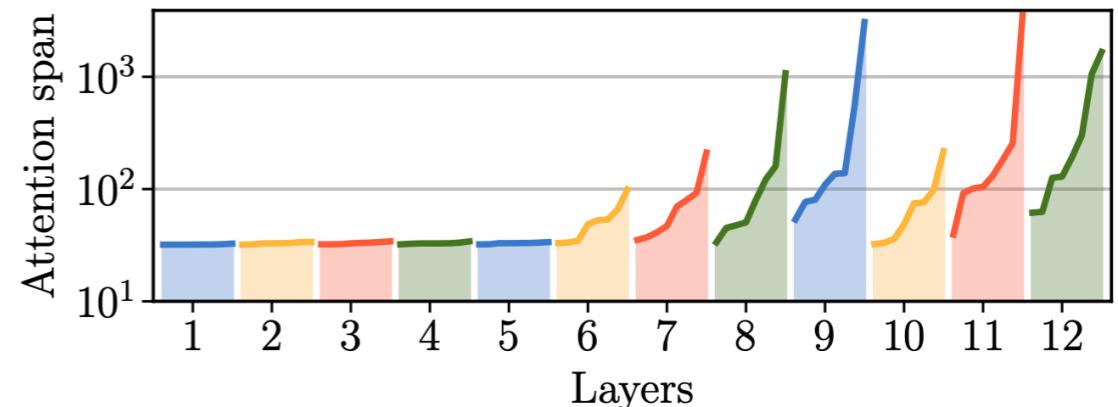
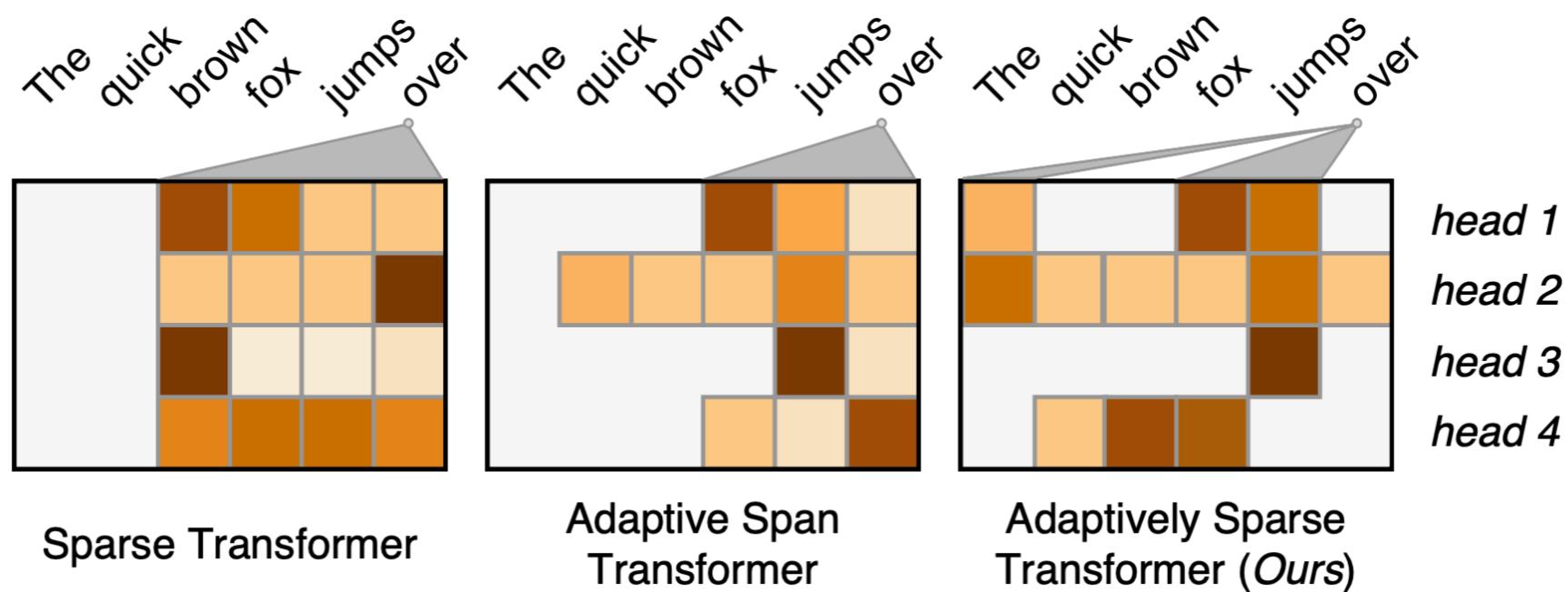


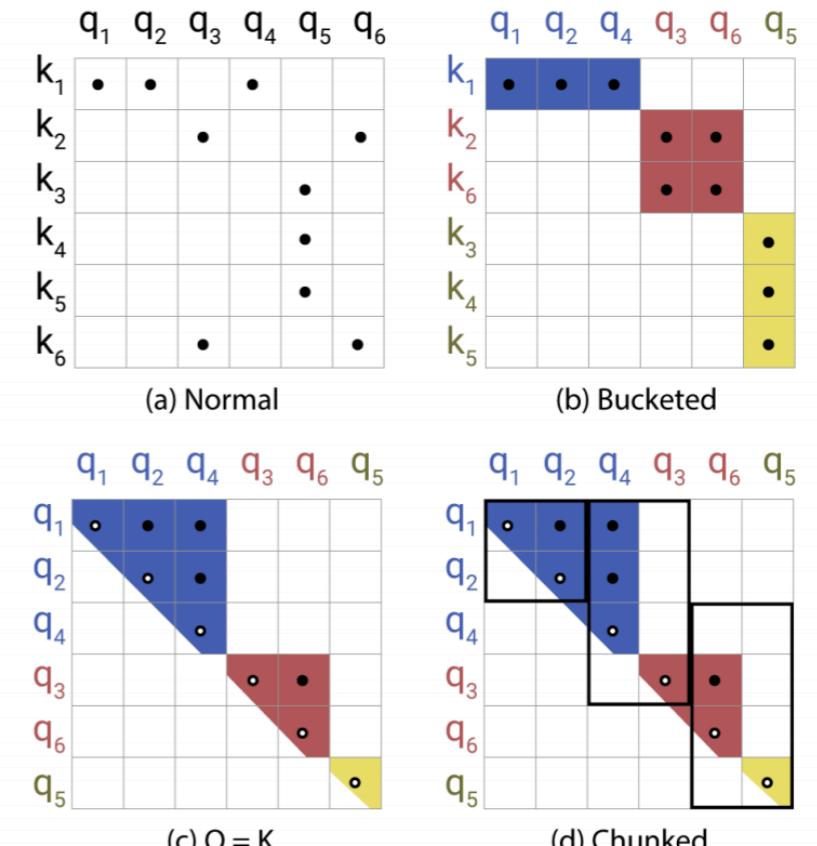
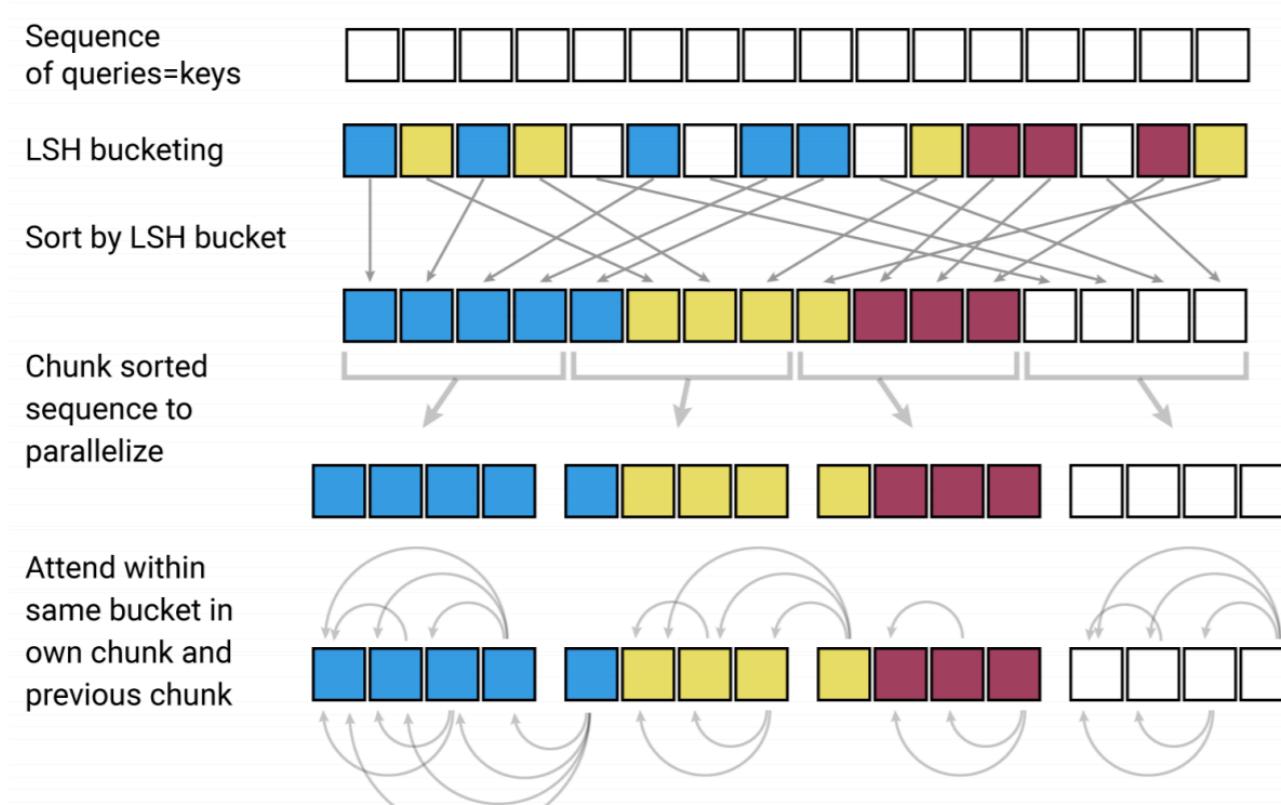
Figure 4: Adaptive spans (in log-scale) of every attention heads in a 12-layer model with span limit  $S = 4096$ . Few attention heads require long attention spans.

- Can be further combined with sparse computation (Correira et al. 2019)



# Reformer: Efficient Adaptively Sparse Attention

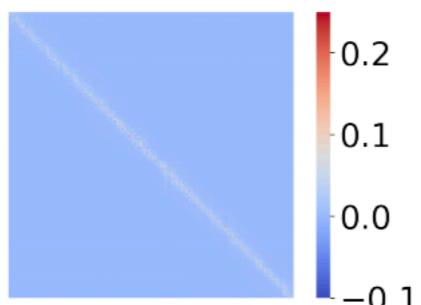
- Chicken-and-egg problem in sparse attention:
  - Can sparsify relatively low-scoring values to improve efficiency
  - Need to calculate all values to know which ones are relatively low-scoring
- **Reformer** (Kitaev et al. 2020): efficient sparsification approximation through
  - Shared key and query parameters to put key and query in the same space
  - Locality sensitive hashing to efficiently calculate high-scoring attention weights
  - Chunking to make sparse computation more GPU friendly



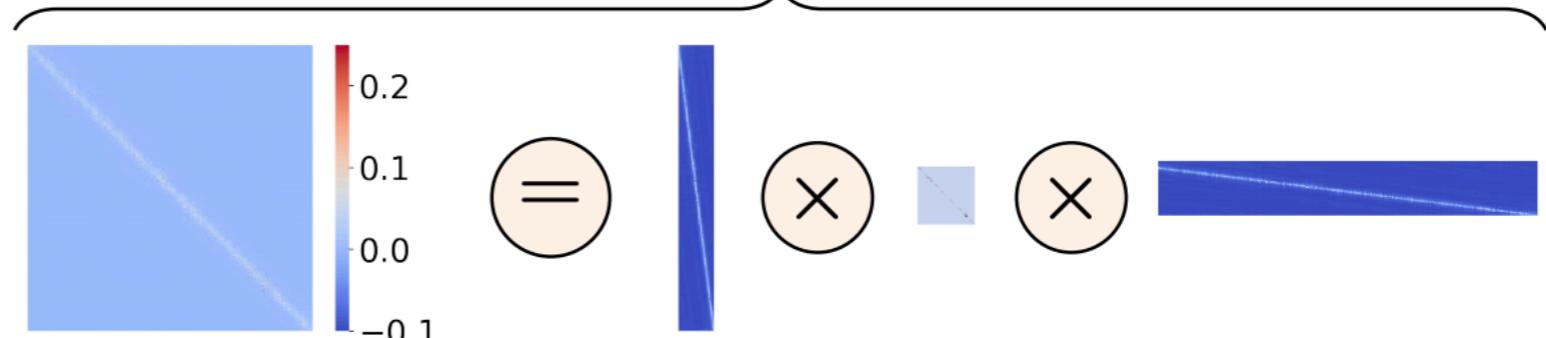
# Low-rank Approximation

- Calculating the attention matrix is expensive, can it be approximated with a low-rank matrix?
- **Linformer:** Add low-rank linear projections into model (Wang et al. 2020)
- **Nystromformer:** Approximate using the Nystrom method, sampling "landmark" points (Xiong et al. 2021)

softmax



Nyström approximation



# How to Evaluate Document-level Models?

- Simple: Perplexity, classification over long documents
- More focused:
  - Sentence scrambling (Barzilay and Lapata 2008)
  - Final sentence prediction (Mostafazadeh et al. 2016)
  - Final word prediction (Paperno et al. 2016)
- Composite benchmark containing several task: Long range arena (Tay et al. 2020)

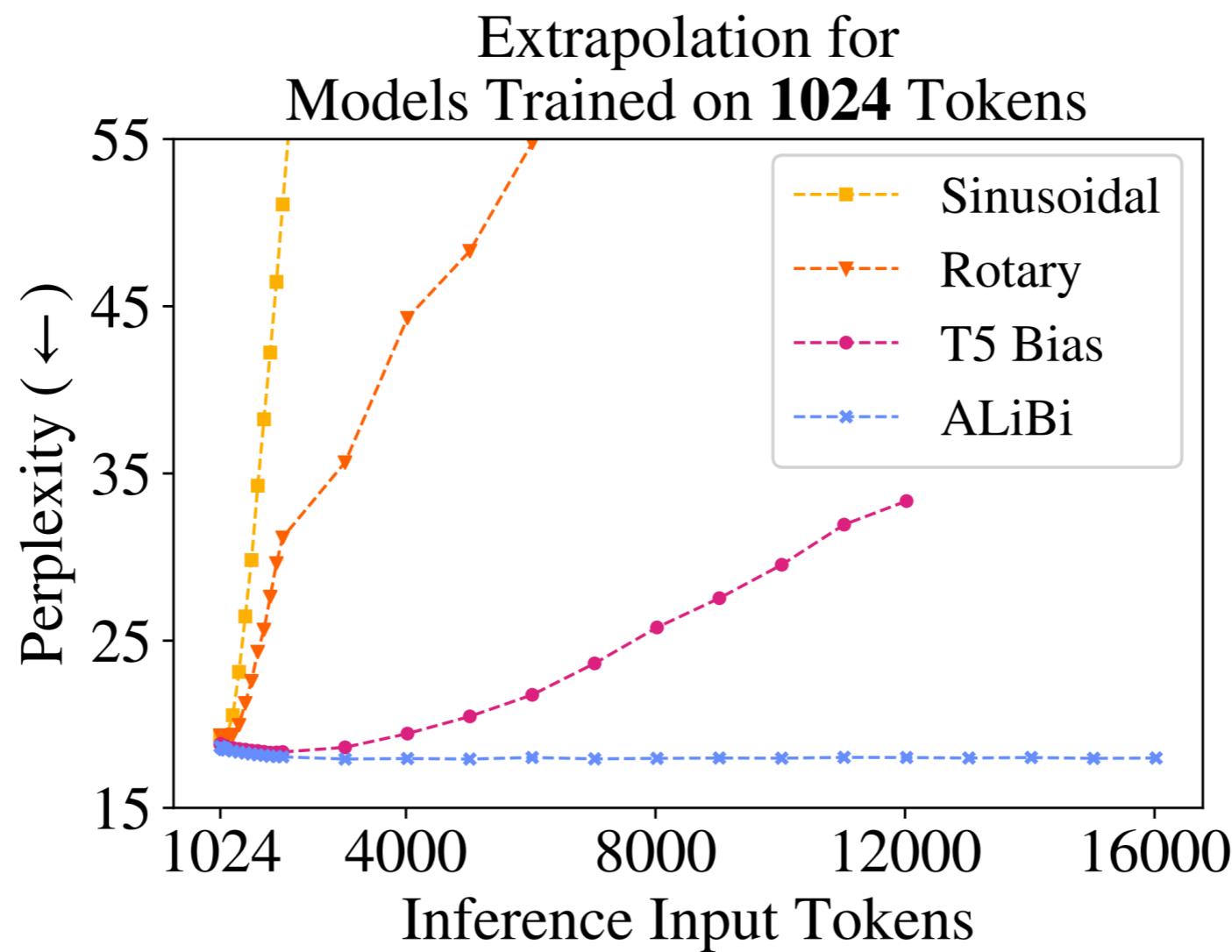
# Generalization Beyond Training Length?

- Key insight: positional embeddings in decoder-only Transformers trained on language data are mostly useful for learning relative distances
- Replace positional embeddings with offset biases in attention, either learned (T5; Raffel et al. 2020) or hard-coded (ALiBi; Press et al. 2022; KERPLE; Chi et al. 2022)

$$\begin{matrix} q_1 \cdot k_1 \\ q_2 \cdot k_1 & q_2 \cdot k_2 \\ q_3 \cdot k_1 & q_3 \cdot k_2 & q_3 \cdot k_3 \\ q_4 \cdot k_1 & q_4 \cdot k_2 & q_4 \cdot k_3 & q_4 \cdot k_4 \\ q_5 \cdot k_1 & q_5 \cdot k_2 & q_5 \cdot k_3 & q_5 \cdot k_4 & q_5 \cdot k_5 \end{matrix} + \begin{matrix} 0 \\ -1 & 0 \\ -2 & -1 & 0 \\ -3 & -2 & -1 & 0 \\ -4 & -3 & -2 & -1 & 0 \end{matrix} \bullet m$$

# Generalization Beyond Training Length?

- Models don't really improve in perplexity when training on longer sequences, but with the right positional encoding, they also don't totally break down



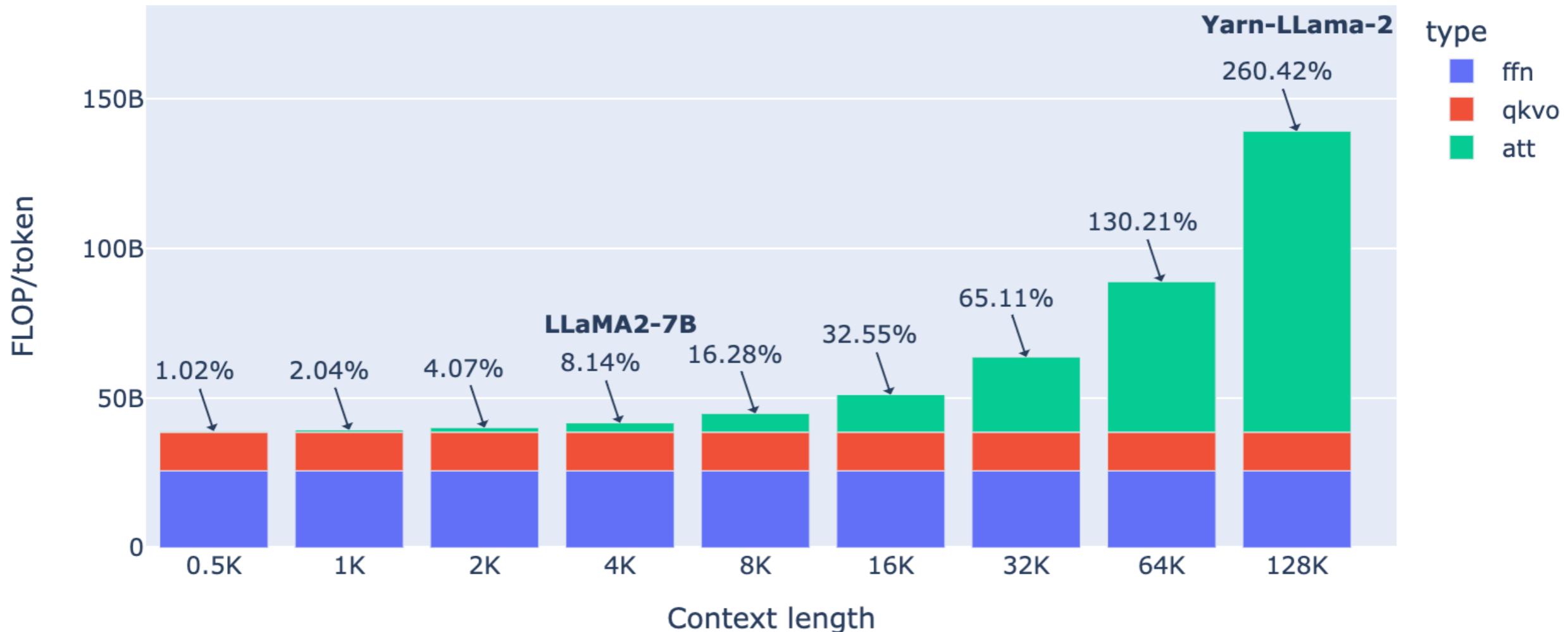
# Computation as a Bottleneck?

1	description	FLOPs / update	% FLOPS MHA	% FLOPS FFN	% FLOPS attn	% FLOPS logit
8	OPT setups					
9	760M	4.3E+15	35%	44%	14.8%	5.8%
10	1.3B	1.3E+16	32%	51%	12.7%	5.0%
11	2.7B	2.5E+16	29%	56%	11.2%	3.3%
12	6.7B	1.1E+17	24%	65%	8.1%	2.4%
13	13B	4.1E+17	22%	69%	6.9%	1.6%
14	30B	9.0E+17	20%	74%	5.3%	1.0%
15	66B	9.5E+17	18%	77%	4.3%	0.6%
16	175B	2.4E+18	17%	80%	3.3%	0.3%
--						

Credit: Stephen Roller from the OPT project;  
<https://twitter.com/stephenroller/status/1579993017234382849/photo/1>

# Computation as a Bottleneck?

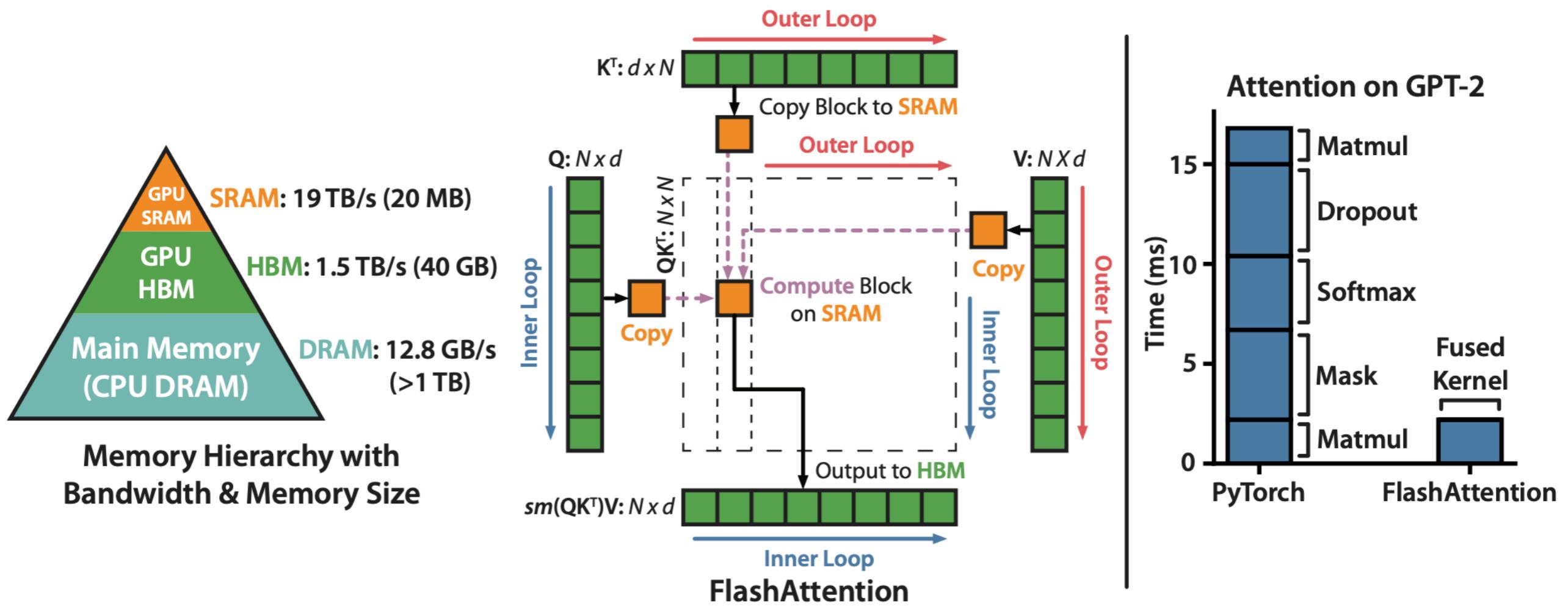
- Even with quadratic scaling of attention, 8-16K tokens is still reasonable on modern hardware



<https://www.harmdevries.com/post/context-length/>

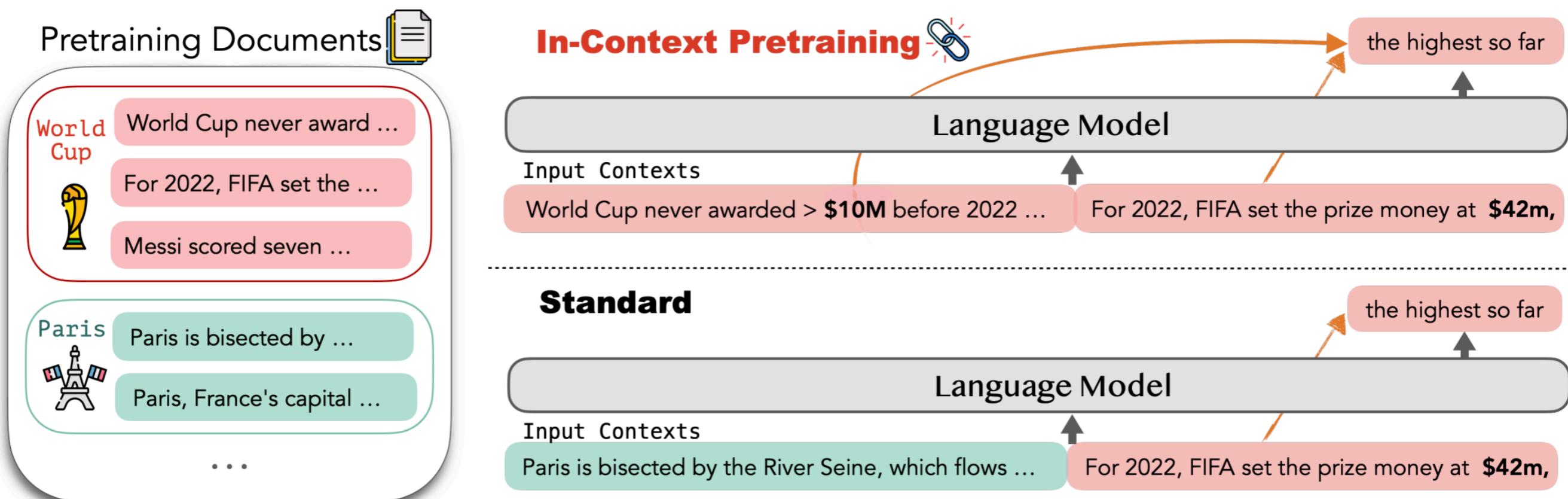
# Memory as a Bottleneck

- Flash Attention (Dao et al. 2022). Still computes exact quadratic attention, but restructure the computation to exploit GPU memory locality



# Data as a Bottleneck

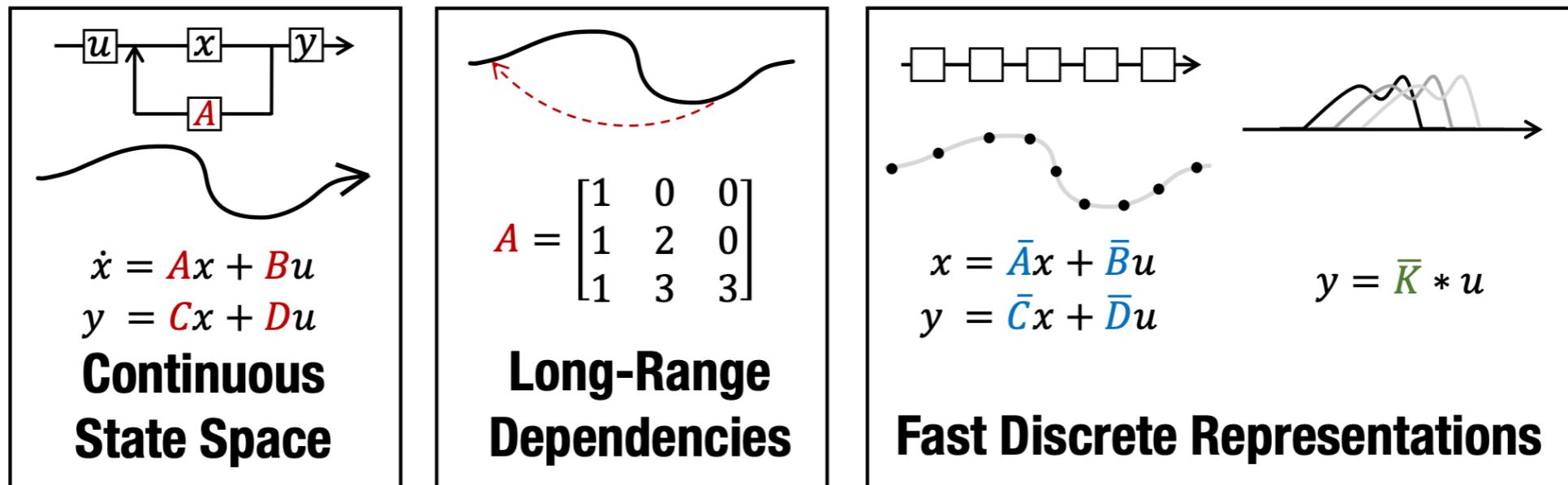
- Concatenating related documents, rather than random documents, improves in-context learning and long-context reasoning (Shi et al. 2023)



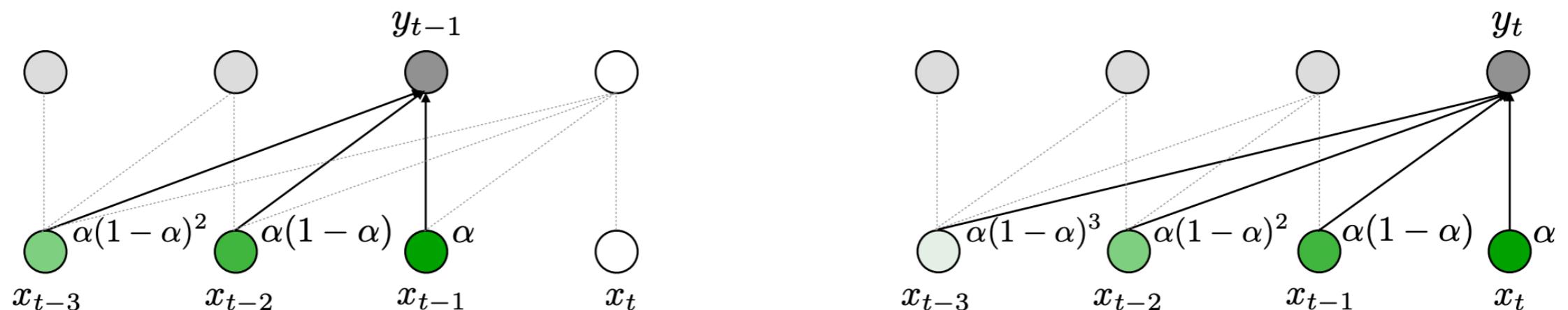
Method	RACE-High	RACE-Middle	BoolQ	SQuAD	HotpotQA	DROP	Average
Standard	39.5	53.3	68.9	26.3	10.5	27.2	37.6
<i>k</i> NN	36.2	51.4	65.3	23.5	14.4	25.1	36.0
ICLM	<b>41.5</b>	<b>56.9</b>	<b>73.0</b>	<b>30.3</b>	<b>21.9</b>	<b>35.7</b>	<b>43.2</b>

# Updates and Averaging

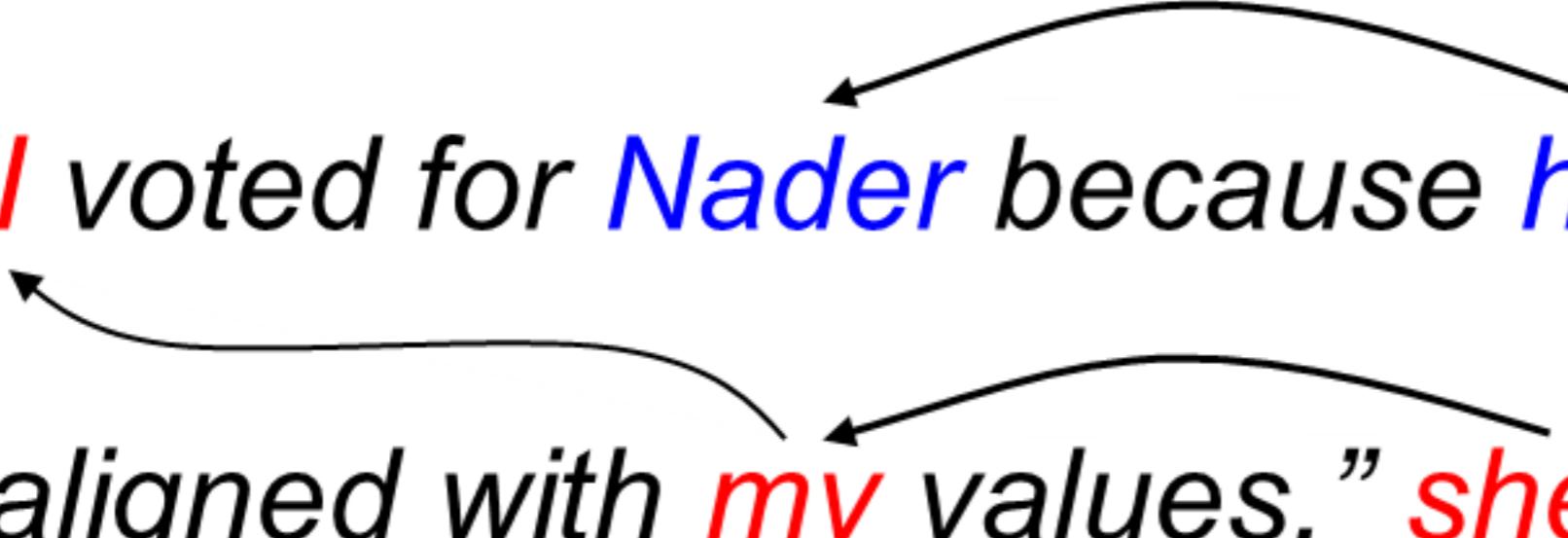
- S4 (Gu et al. 2022): A linear recurrence expands into something like  $Ax_t + A^2x_{t-1} + A^3x_{t-2} + \dots$



- MEGA (Ma et al. 2023): use linear decay



*“I voted for Nader because he was most aligned with my values,” she said.*



## Entity Coreference

# Document Problems: Entity Coreference

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch.

A renowned speech therapist was summoned to help the King overcome his speech impediment...

Example from Ng, 2016

- Step 1: Identify Noun Phrases mentioning an entity (note the difference from named entity recognition).
- Step 2: Cluster noun phrases (**mentions**) referring to the same underlying world **entity**.

# Mention(Noun Phrase) Detection

A renowned speech therapist was summoned to help the King overcome his speech impediment...

A renowned speech therapist was summoned to help the King overcome his speech impediment...

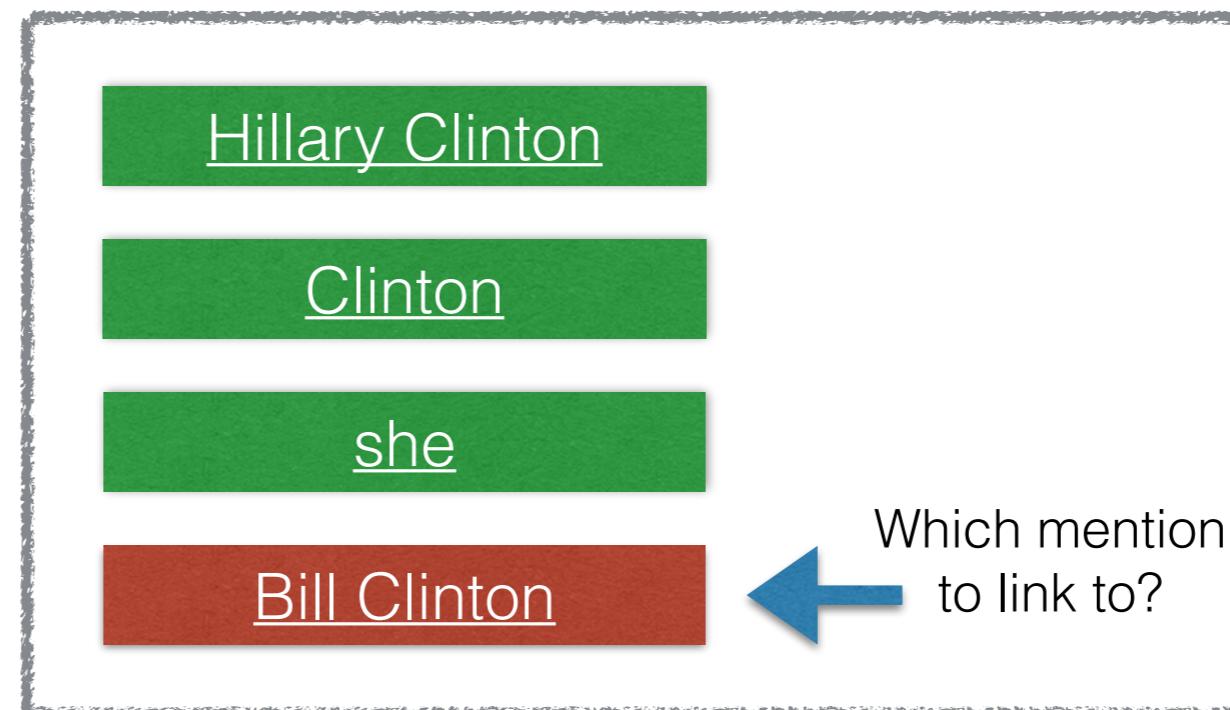
- One may think coreference is simply a clustering problem of given Noun Phrases.
  - Detecting relevant noun phrases is a difficult and important step.
  - Knowing the correct noun phrases affect the result a lot.
  - Normally done as a preprocessing step.

# Components of a Coreference Model

- Like a traditional machine learning model:
  - We need to know the **output space** (e.g. shift-reduce operations in parsing).
  - We need to design the **features**.
  - We need to optimize towards the **evaluation metrics**.
  - **Search algorithm** for structure

# Coreference Models: Output Spaces

- Coreference is a structured prediction problem:
  - Possible cluster structures are in exponential number of the number of mentions. (Number of partitions)
- Models are designed to approximate/explore the space, the core difference is the way each instance is constructed:
  - Mention-based
  - Entity-based



# Mention Pair Models

- The simplest one: Mention Pair Model:
  - Classify the coreference relation between every 2 mentions.
- Simple but many drawbacks:
  - May result in conflicts in transitivity.
  - Too many negative training instances.
  - Do not capture **entity/cluster level** features.
  - No ranking of instances.

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch.  
A renowned speech therapist was summoned to help the King overcome his speech impediment...

✓: Queen Elizabeth <-> her  
✗: Queen Elizabeth <-> husband  
✗: Queen Elizabeth <-> King George VI  
✗: Queen Elizabeth <-> a viable monarch

.....

# Entity Models: Entity-Mention Models

- Entity-Mention Models
  - Create an instance between a mention and a previous\* cluster.

Daume & Marcu (2005);  
Cullotta et al. (2007)

## Example Cluster Level Features:

- Are the genders all compatible?
- Is the cluster containing pronouns only?
- Most of the entities are the same gender?????
- Size of the clusters?

## Problems:

- No ranking between the antecedents.
- Cluster level features are difficult to design.

\* This process often follows the natural discourse order, so we can refer to partially built clusters.

# Advantages of Neural Network Models for Coreference

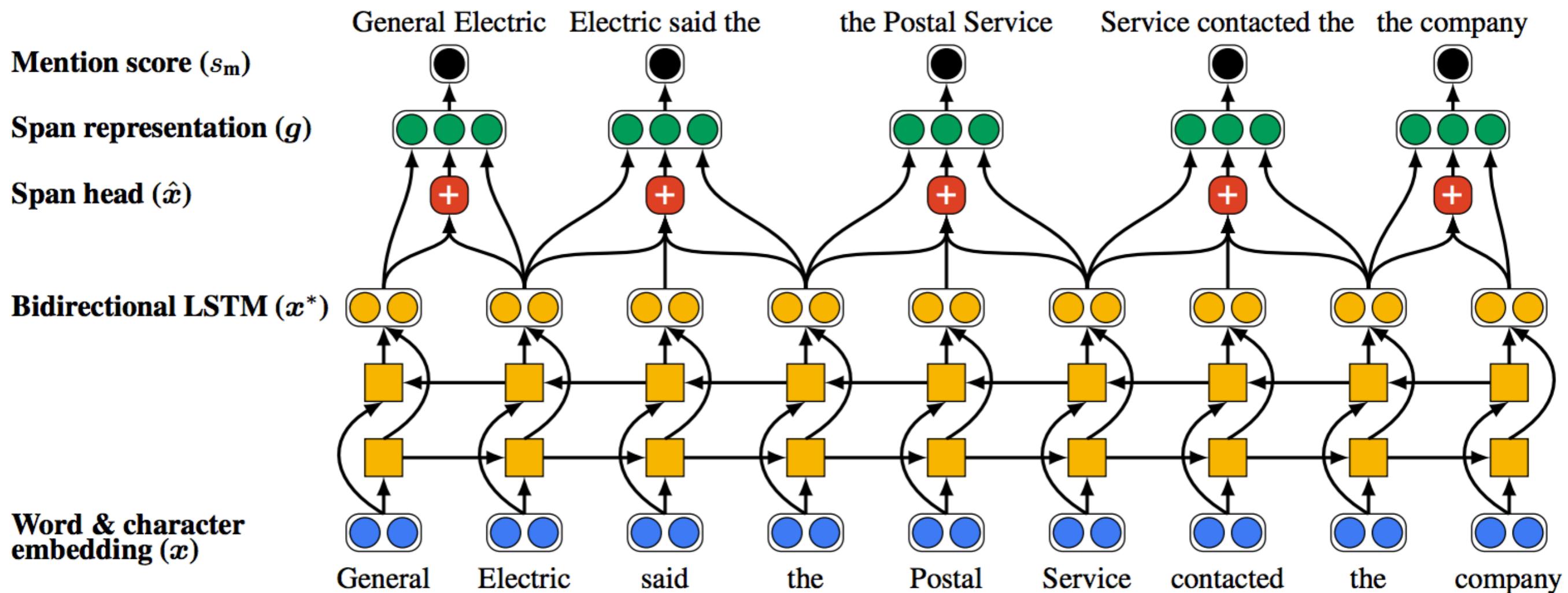
- **Learn the features** with embeddings since most of them can be captured by surface features.
- **Train towards the metric** using reinforcement learning or margin-based methods.
- **Jointly perform mention detection** and clustering.

# End-to-End Neural Coreference

Lee et.al (2017)

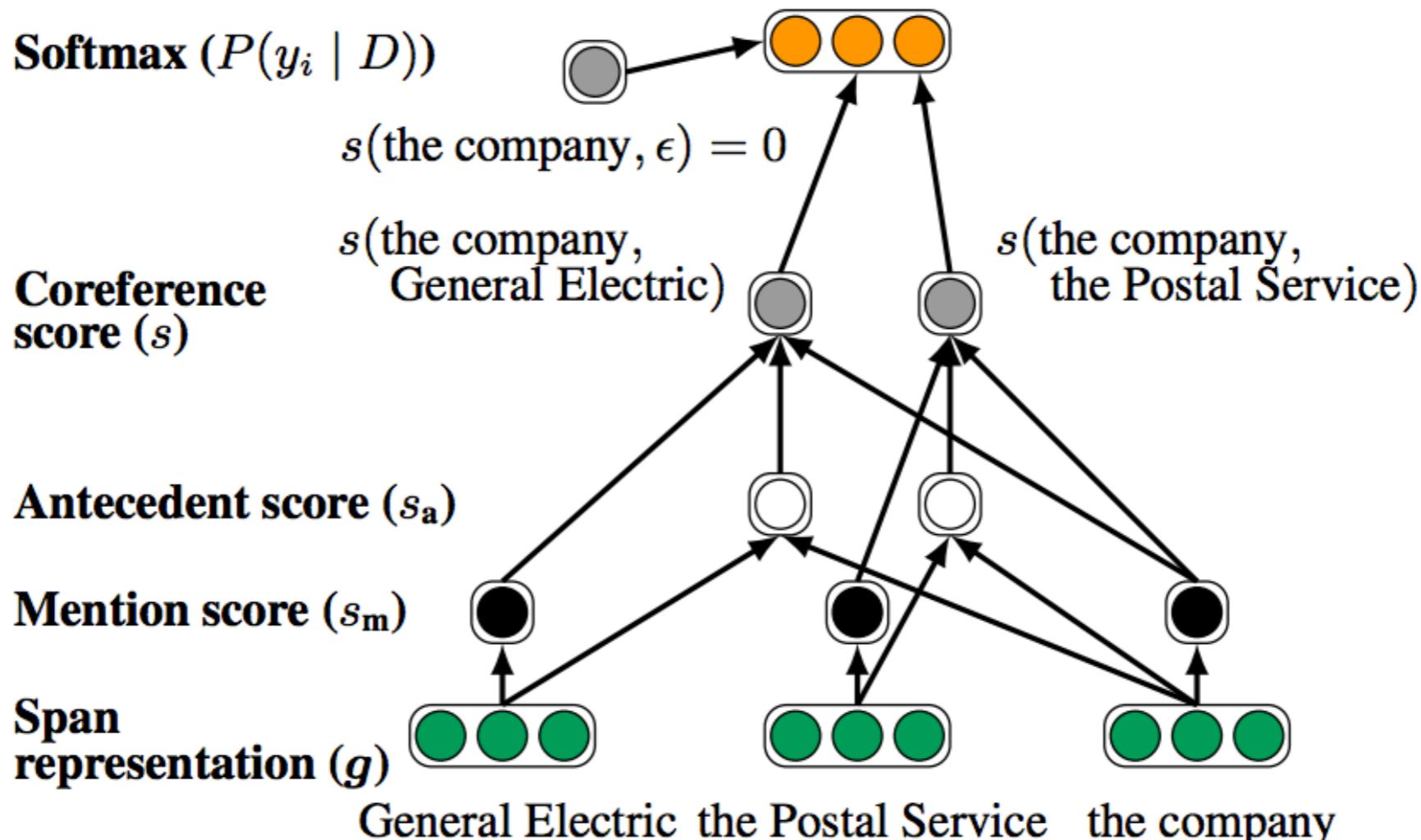
- 2 main contributions of the paper:
  - Can we represent all features with a typical neural network based-embedding?
  - Can neural network allow errors to flow end-to-end? All the way to mention detection?

# End-to-End Neural Coreference (Span Model)



- Build mention representation from word representation (all possible spans)
- Head extracted by self-attention.

# End-to-End Neural Coreference (Coreference Model)

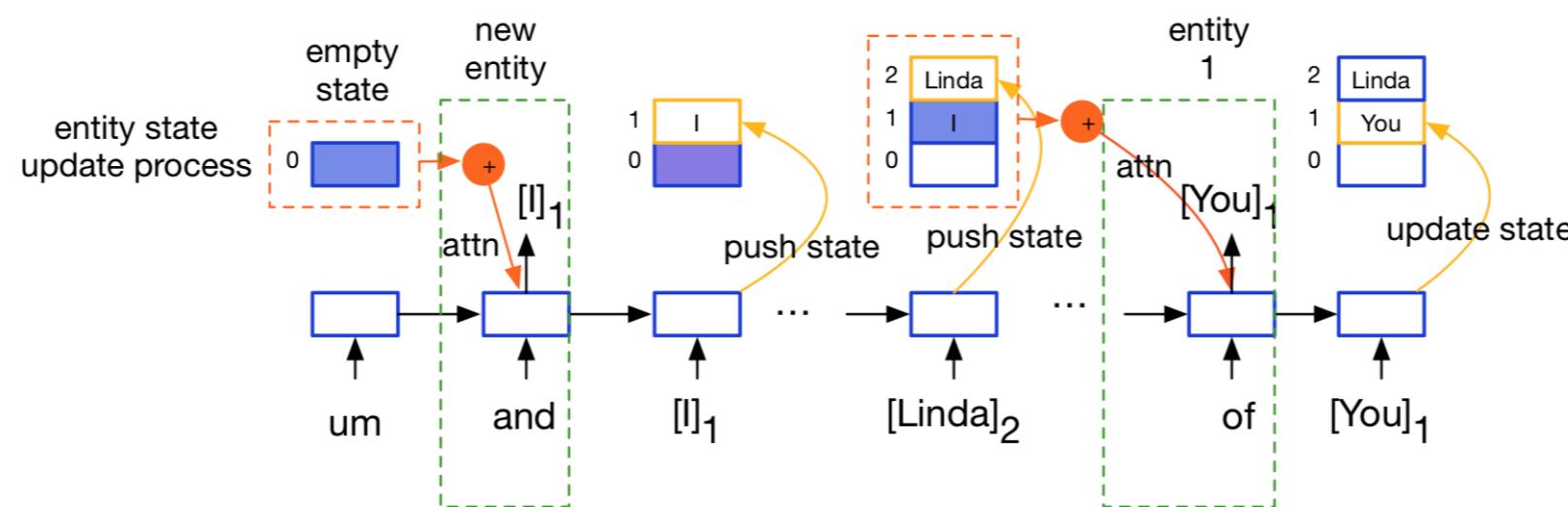


- Coreference model is similar to a mention ranking.
- Coreference score consist of multiple scores.
- Simple max-likelihood

# Using Coreference in Neural Models

- Co-reference aware language modeling (Yang et al. 2017)

um and [I]<sub>1</sub> think that is whats - Go ahead [Linda]<sub>2</sub>. Well and thanks goes to [you]<sub>1</sub> and to [the media]<sub>3</sub> to help [us]<sub>4</sub>...So [our]<sub>4</sub> hat is off to all of [you]<sub>5</sub>...



- Co-reference aware QA models (Dhingra et al. 2017)

mary — got — the — football — she — went — to — the — kitchen — she — left — the — ball — there

# Questions?