CS769 Advanced NLP

Modeling Long Sequences

(Document-Level Models)

Junjie Hu

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Slides adapted from Zhengzhong, Graham

https://junjiehu.github.io/cs769-spring23/

Goal for Today

1. Document-level Neural Language Modeling
    - RNN-based Models
    - Transformer-based Models
2. Other Document-level Tasks
    - Entity Coreference
    - Discourse Parsing

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?'

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

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

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

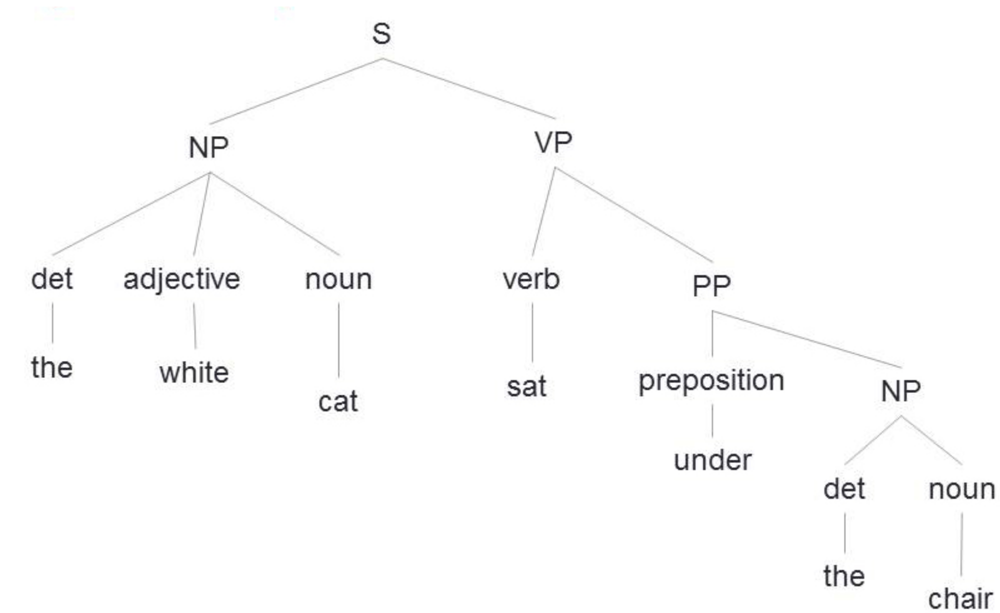
$P(w_{i+1}=\text{beginning} \mid w_i=\text{was}) = 1/2$

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

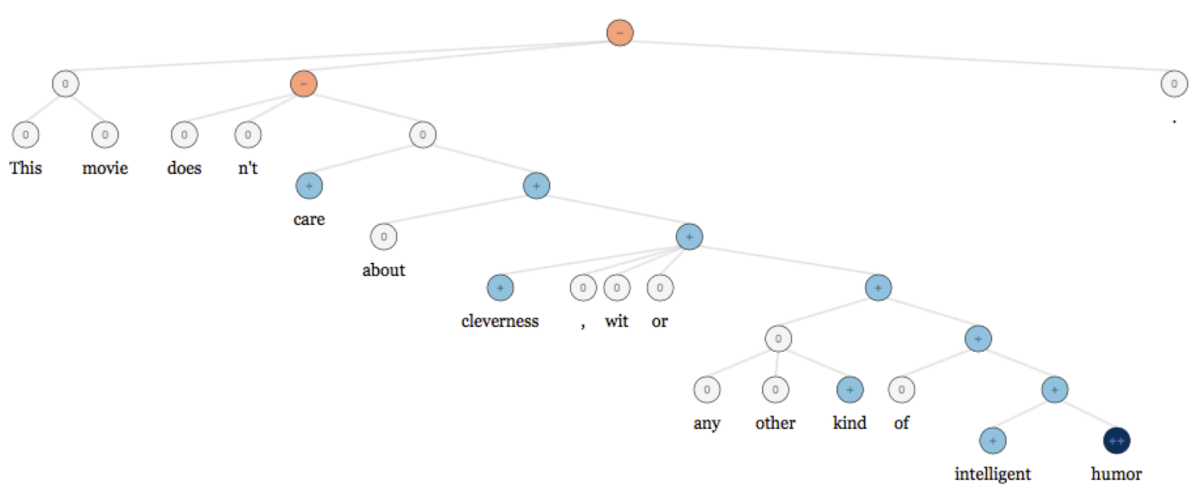
$P(w_{i+1}=\text{book} \mid w_i=\text{the}) = 1/3$

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

Language Models



Parsing



Classification

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?'

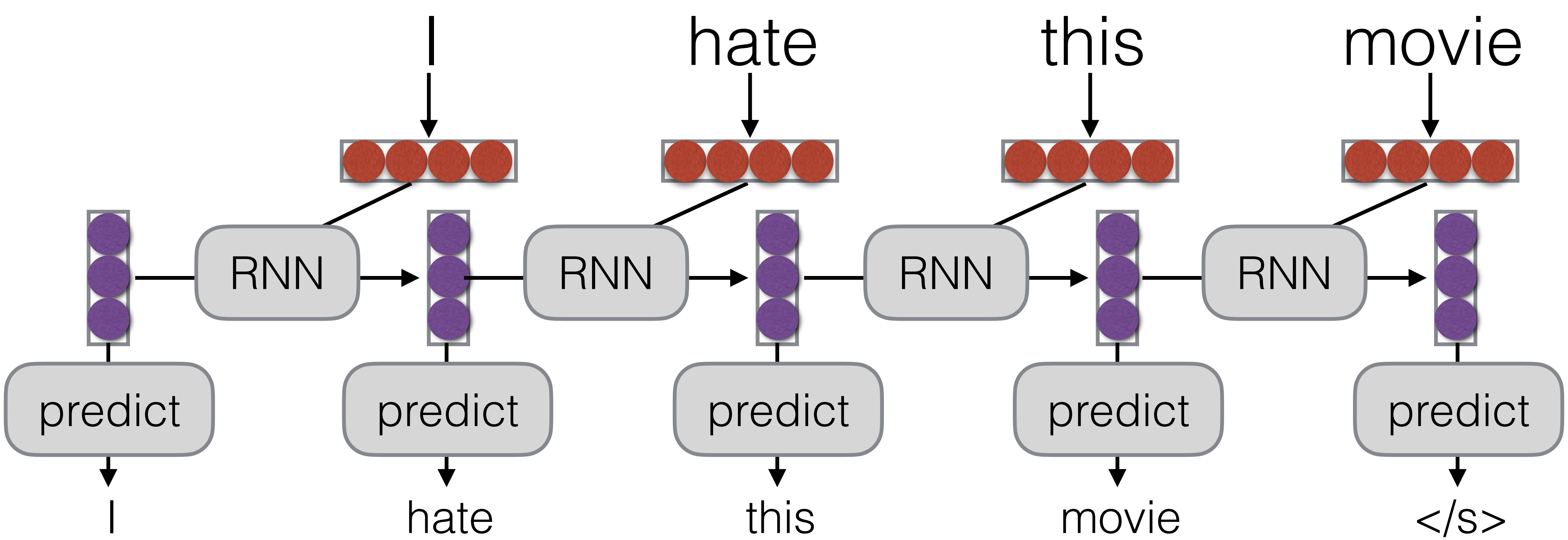
Document Level Language Modeling

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.

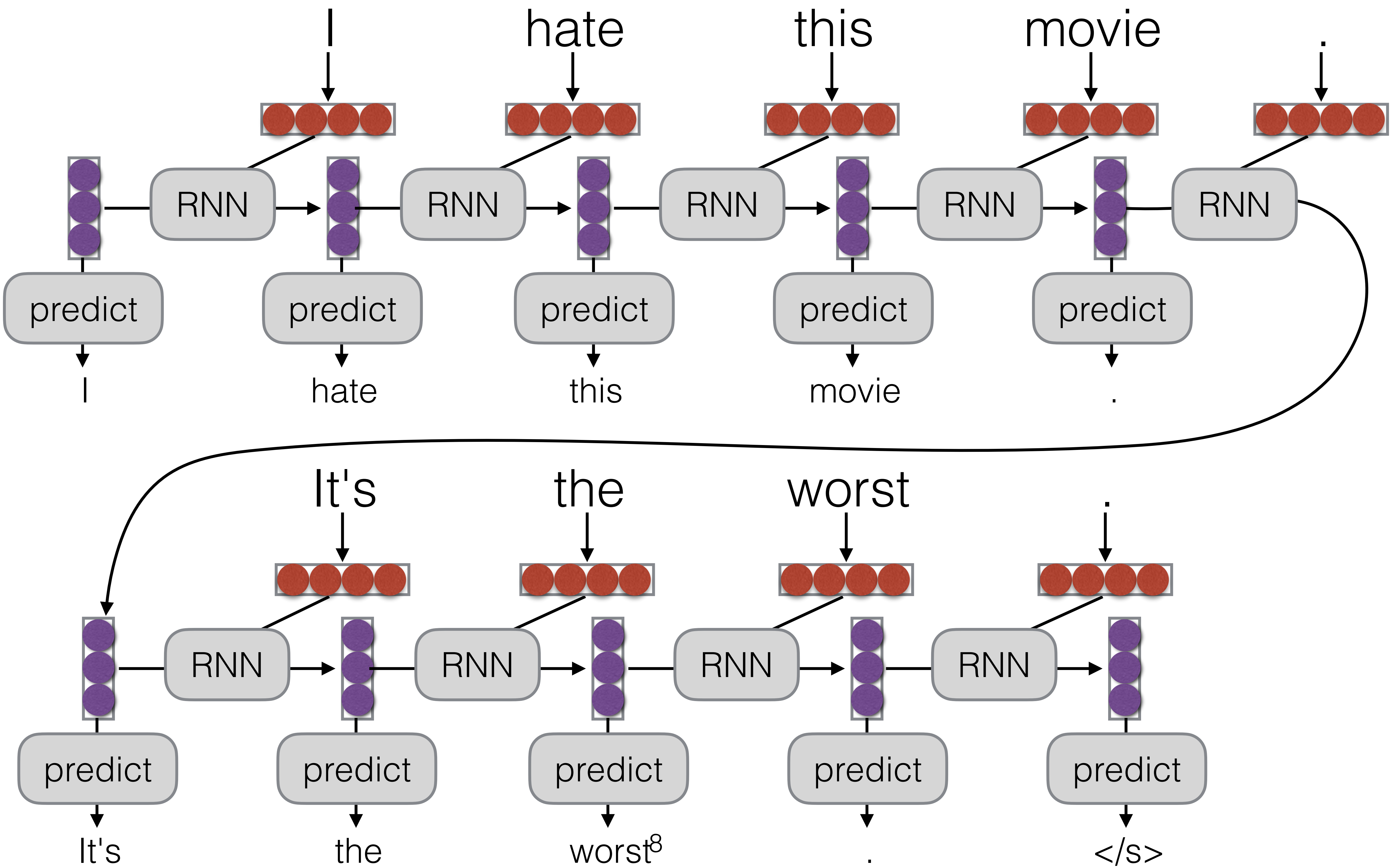
Remember: Modeling using Recurrent Networks

- Model passing previous information in hidden state



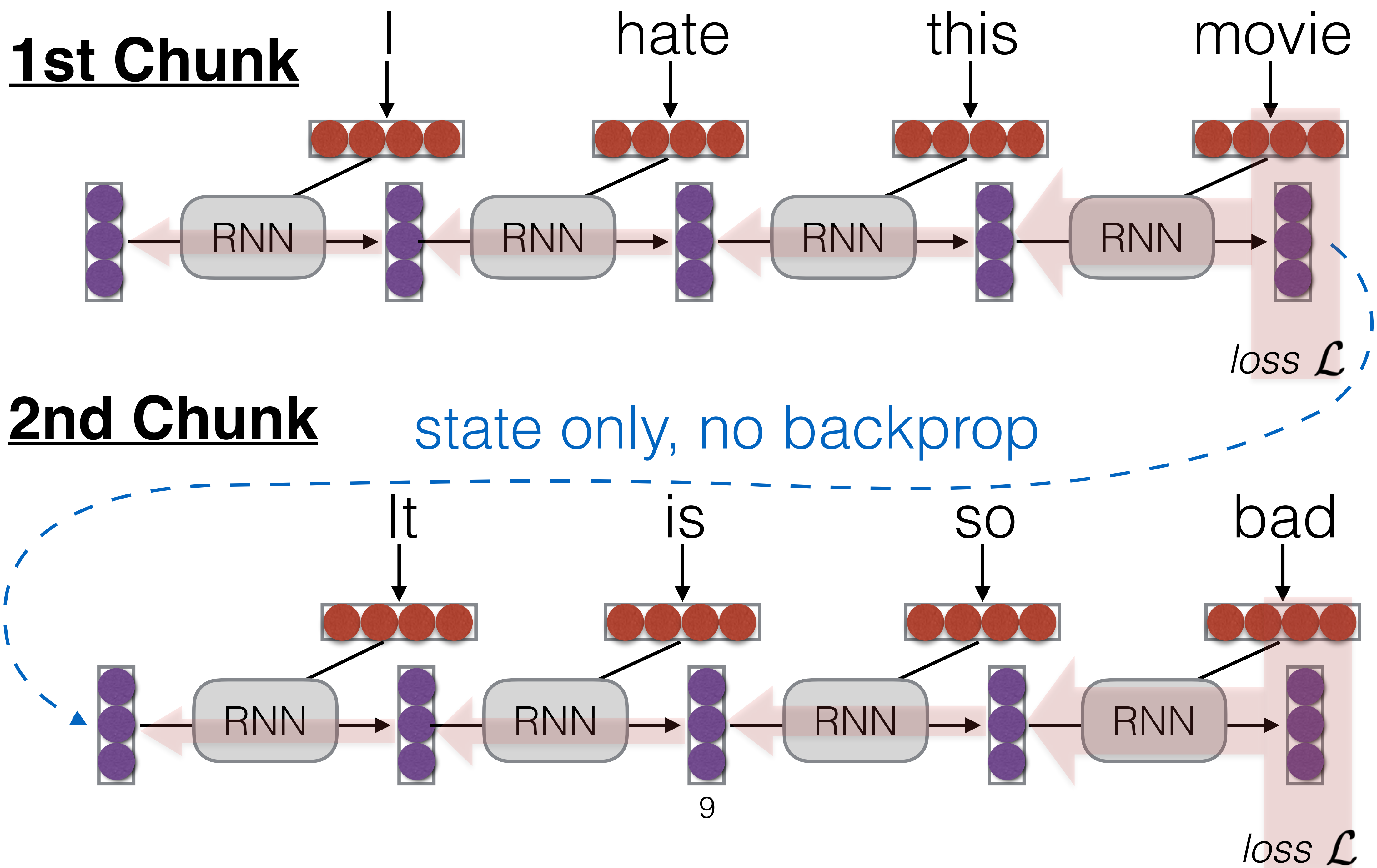
Simple: Infinitely Pass State by RNN LM

(Mikolov et al. 2011)



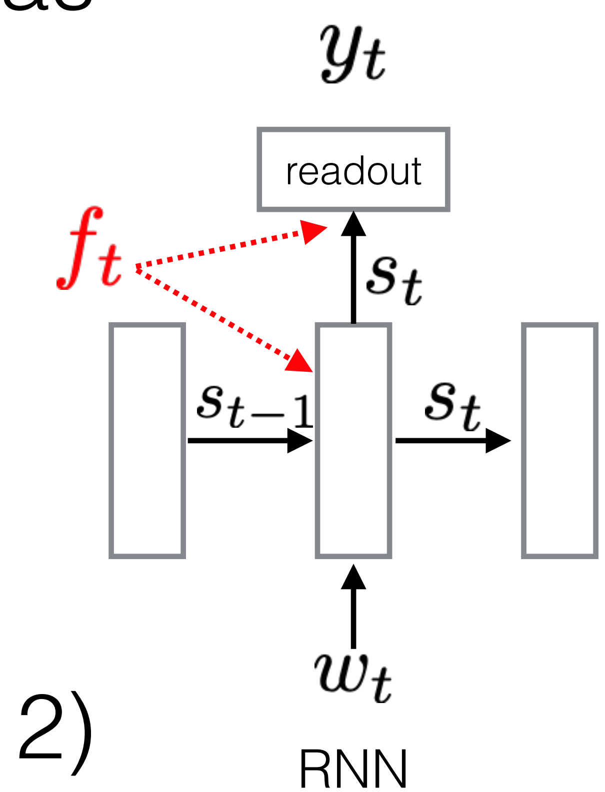
Truncated Backpropagation Through Time (TBPTT) (Elman 1990, Boden 2001)

- The backpropagation update is performed back for a fixed number of past time steps.



Separate Encoding for Coarse-grained Document Context

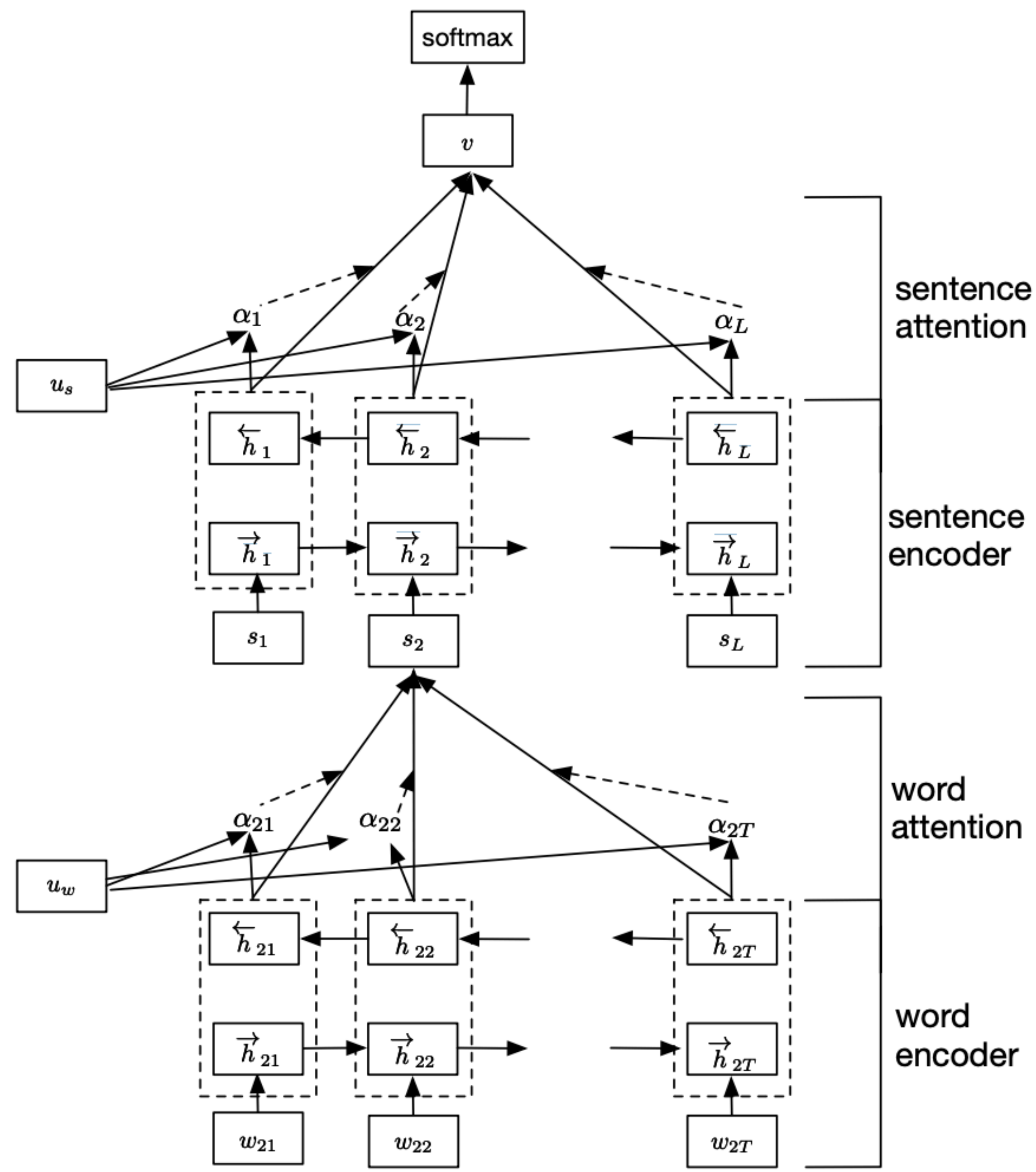
- Explicitly add the external global features f_t as input to
  1. each RNN cell
  2. The final readout linear layer
- What global context?
  - Use topic modeling (Mikolov & Zweig 2012)
  - Use bag-of-words of previous sentence(s), optionally with attention (Wang and Cho 2016)
  - Use last state of previous sentence (Ji et al. 2015)



Hierarchical Attention Network

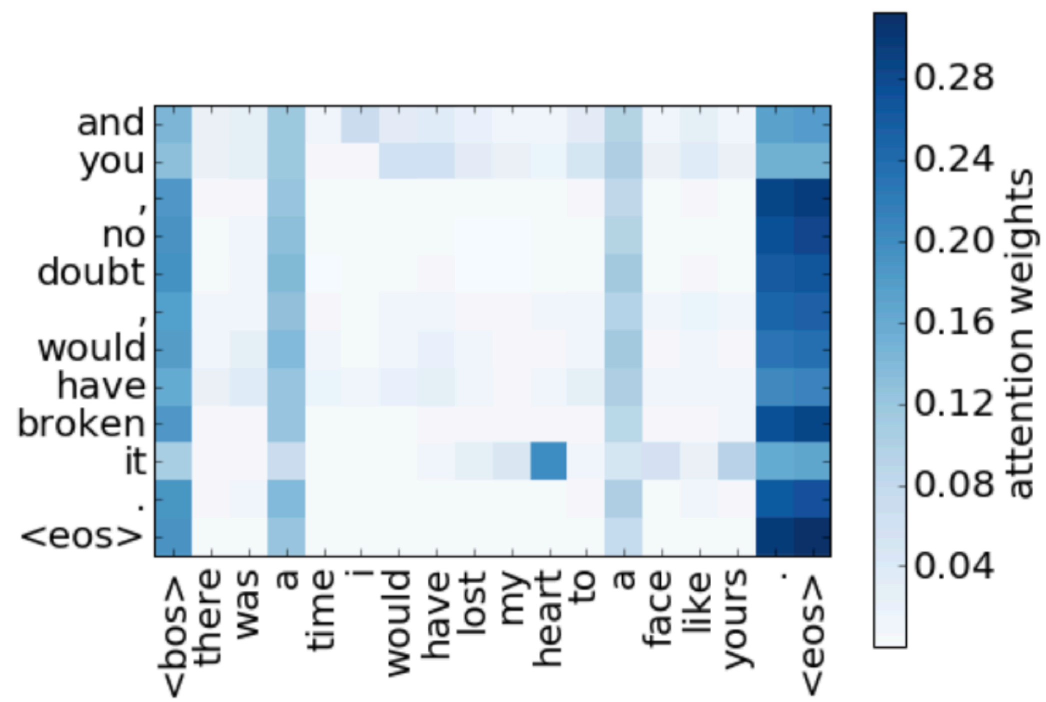
(Yang et al. 2018)

- One word-level BiGRU to encode words within a sentence
- Learn a weighted sum of word hidden vectors as the sentence representation.
- One sentence-level BiGRU to encode sentences within a document
- Weighted sum of sentence hidden vectors as the doc representation.



Transformers Across Sentences

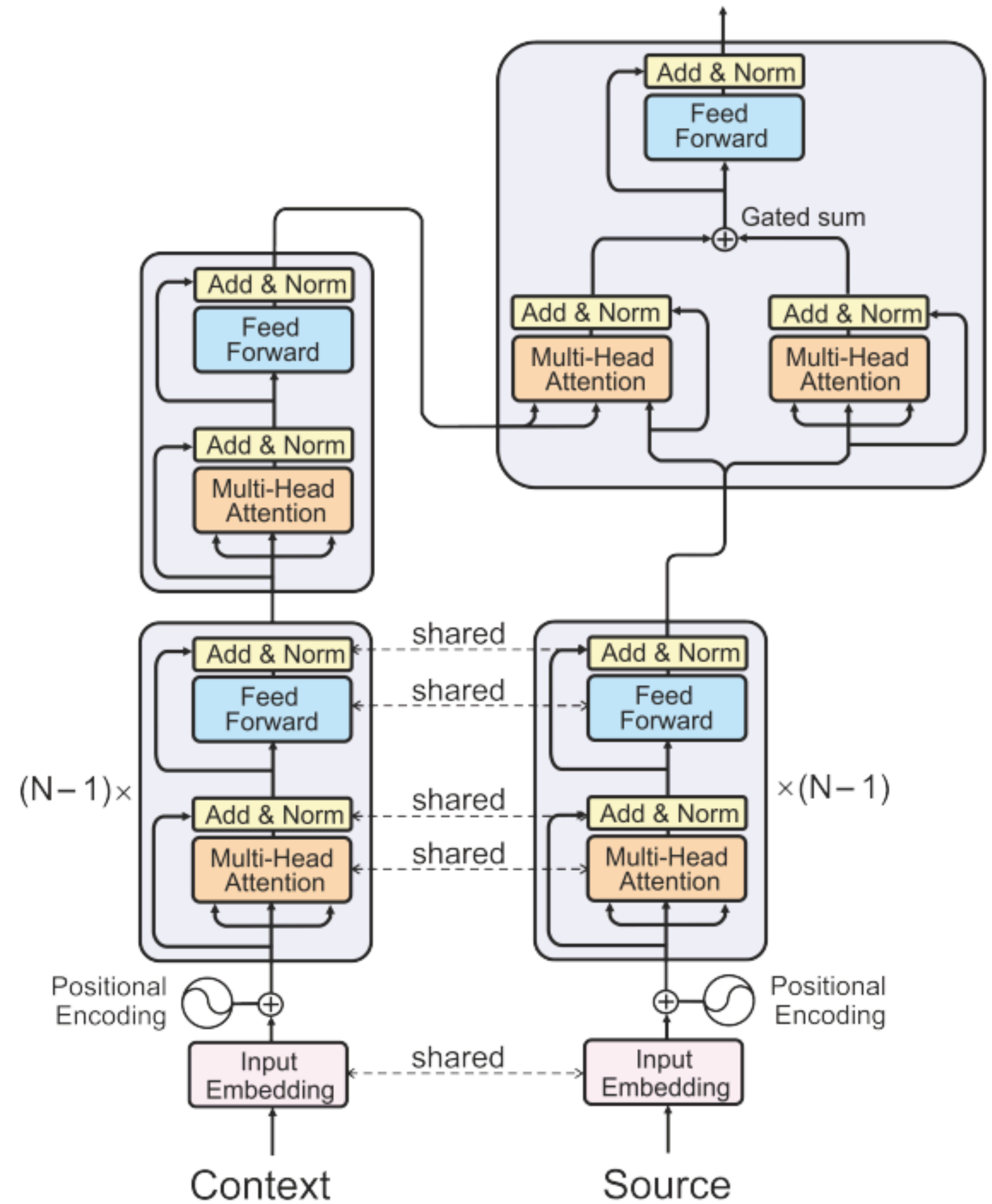
- Simply self-attend to all words in the document
  - + Can simply use document-level context
  - + Can learn interesting phenomena (e.g. co-reference)



- - Computation of the attention matrix is quadratic in sequence length $O(L^2)$!

Encode Context and Source Separately (Elena et al. 2018)

- Use two Transformer encoders to encode the **context and current source sentence separately** instead of a combined document.
- Share the first N-1 layers for the two encoders.
- Context: previous/next sentence, or random sentence in the doc
- + Reduce the computation from quadratic of **doc length** $O(L^2)$ to quadratic of **sentence length** $O(l^2)$

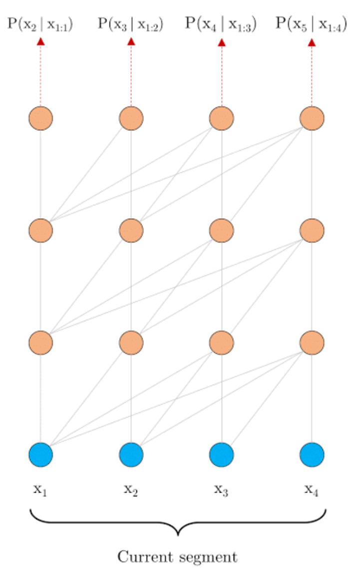


Transformer-XL: Truncated BPTT+Transformer

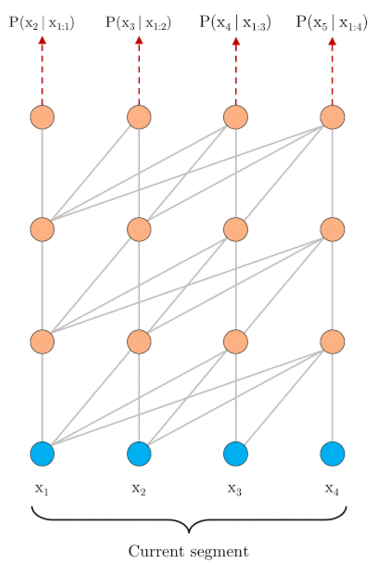
(Dai et al. 2019)

- Standard Transformer: encode each chunk separately
- Transformer-XL: attend to fixed **vectors** from the previous sentence

Standard Transformer



Transformer-XL

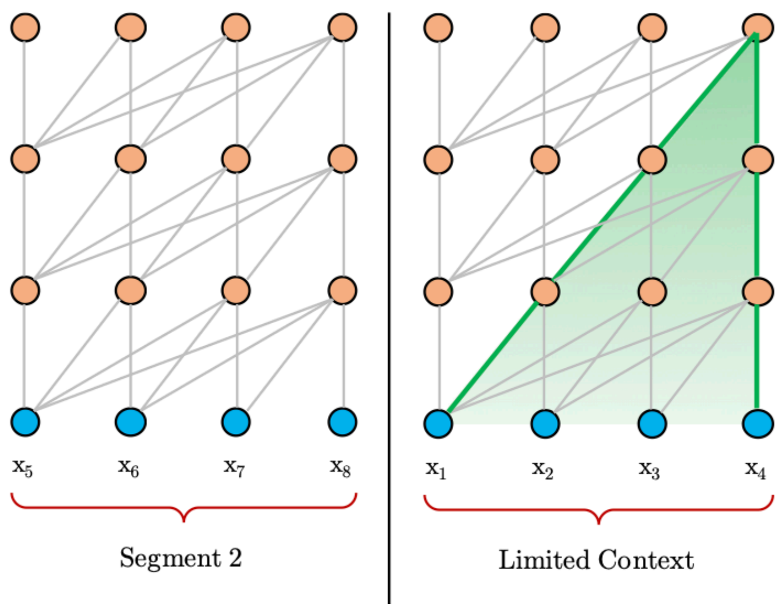


Transformer-XL: Truncated BPTT+Transformer

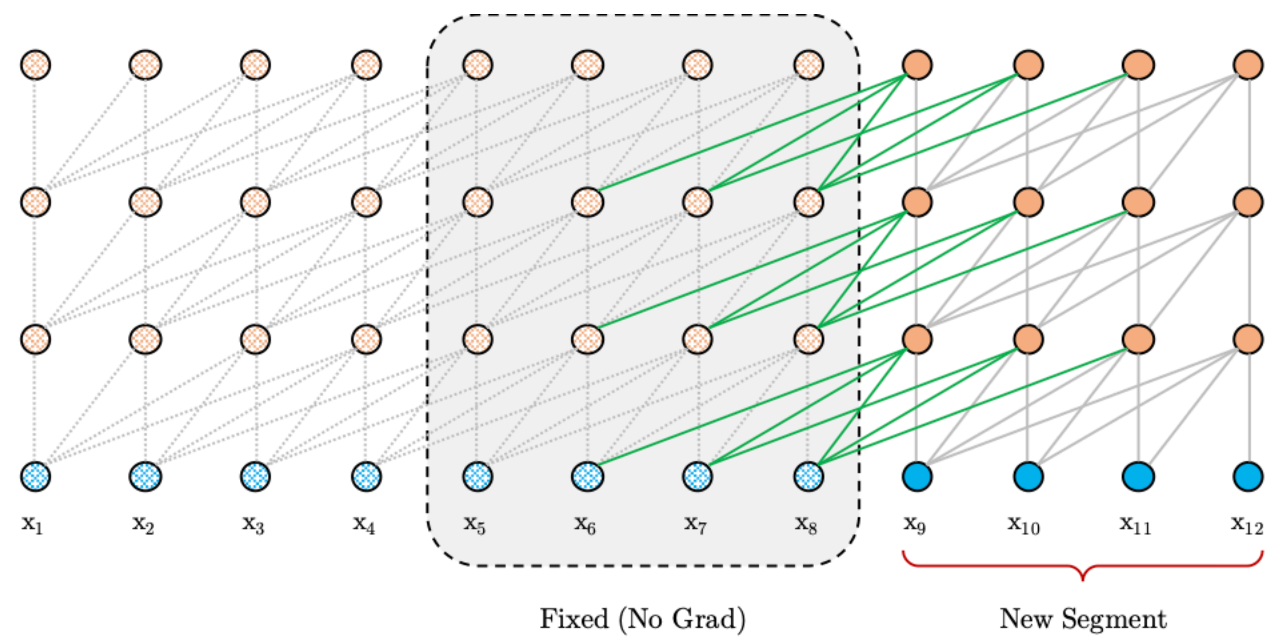
(Dai et al. 2019)

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

Standard Transformer



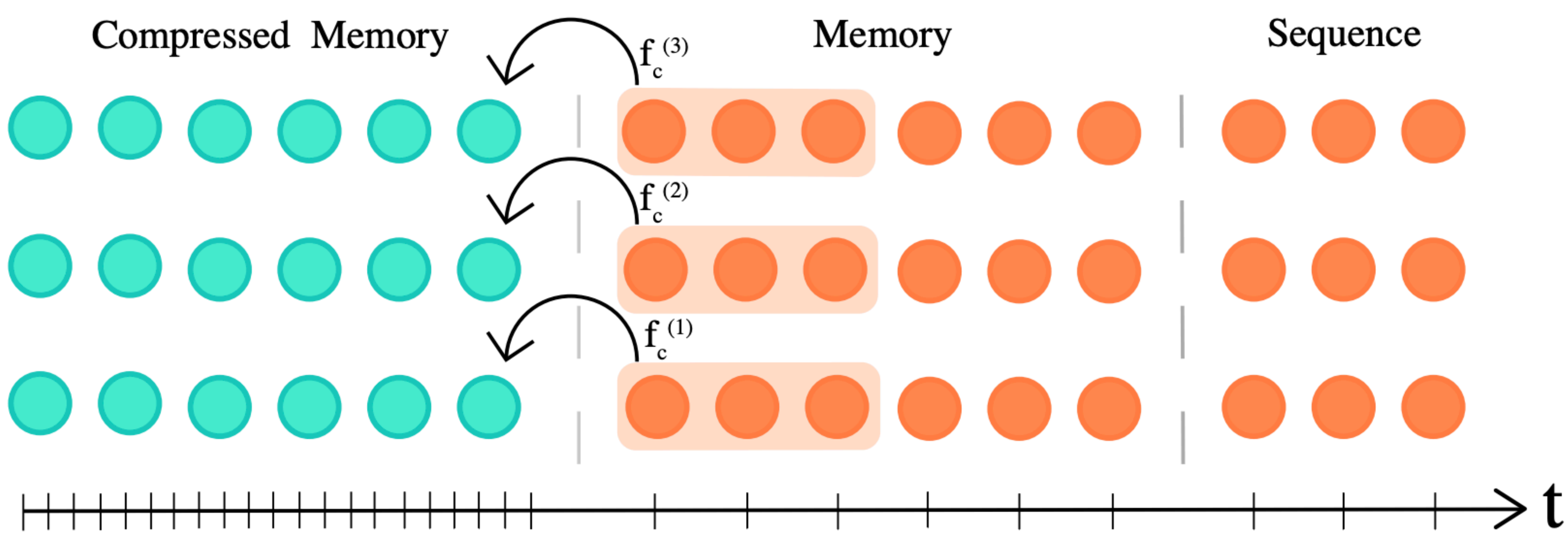
Transformer-XL



- How far away can Transformer-XL look back?
    - $O(N \times l)$, N is the no. of layers, l is the no. of words in a chunk

Compressing Previous States

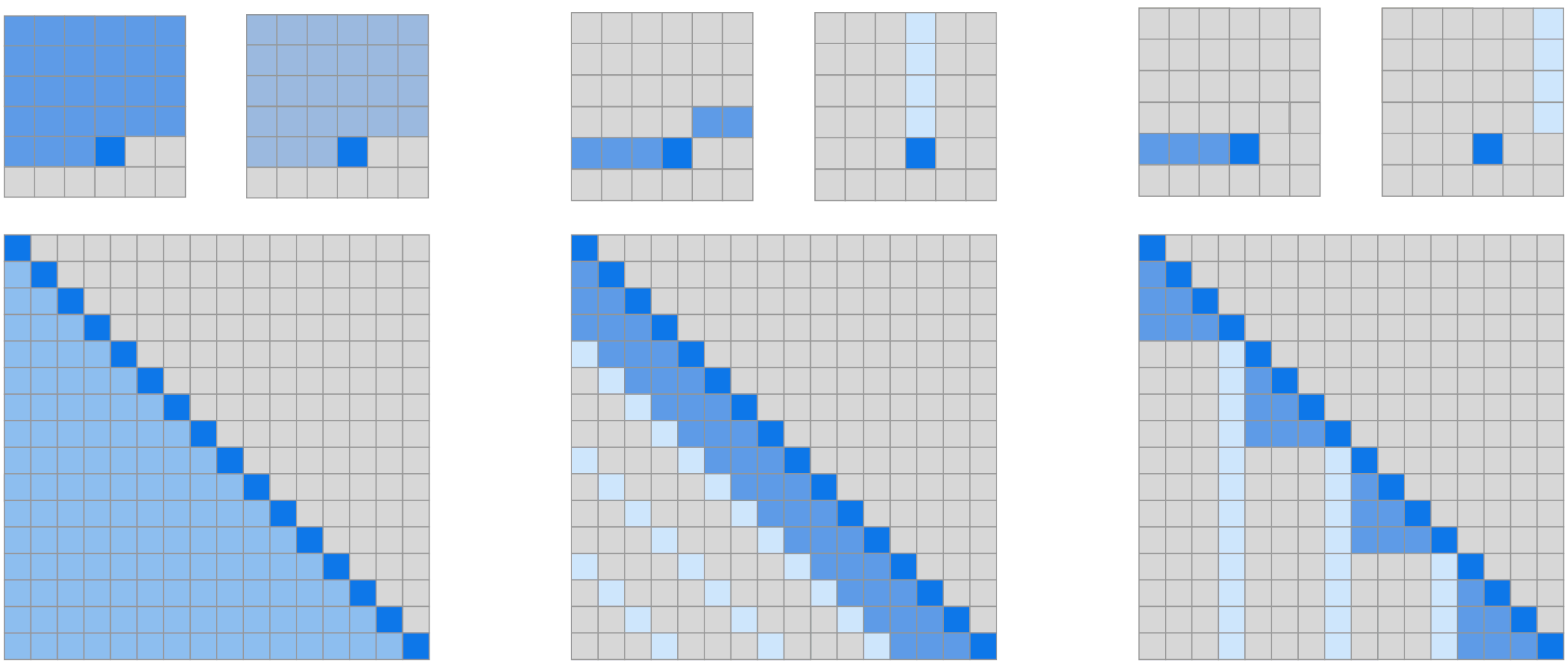
- Extension over Transformer-XL that uses the last chunk as the memory
- Add a "strided" compression step over previous states (Rae & Potapenko et al. 2019)



Sparse Transformers

(Child et al. 2019)

- Add "stride", only attending to every *n* previous states



(a) Transformer (b) Sparse Transformer (strided) (c) Sparse Transformer (fixed)

Adaptive Span Transformers

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

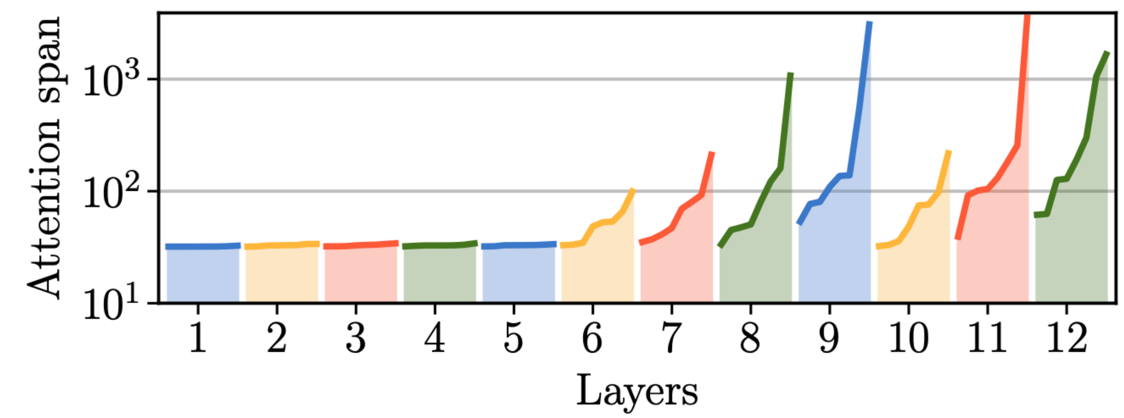
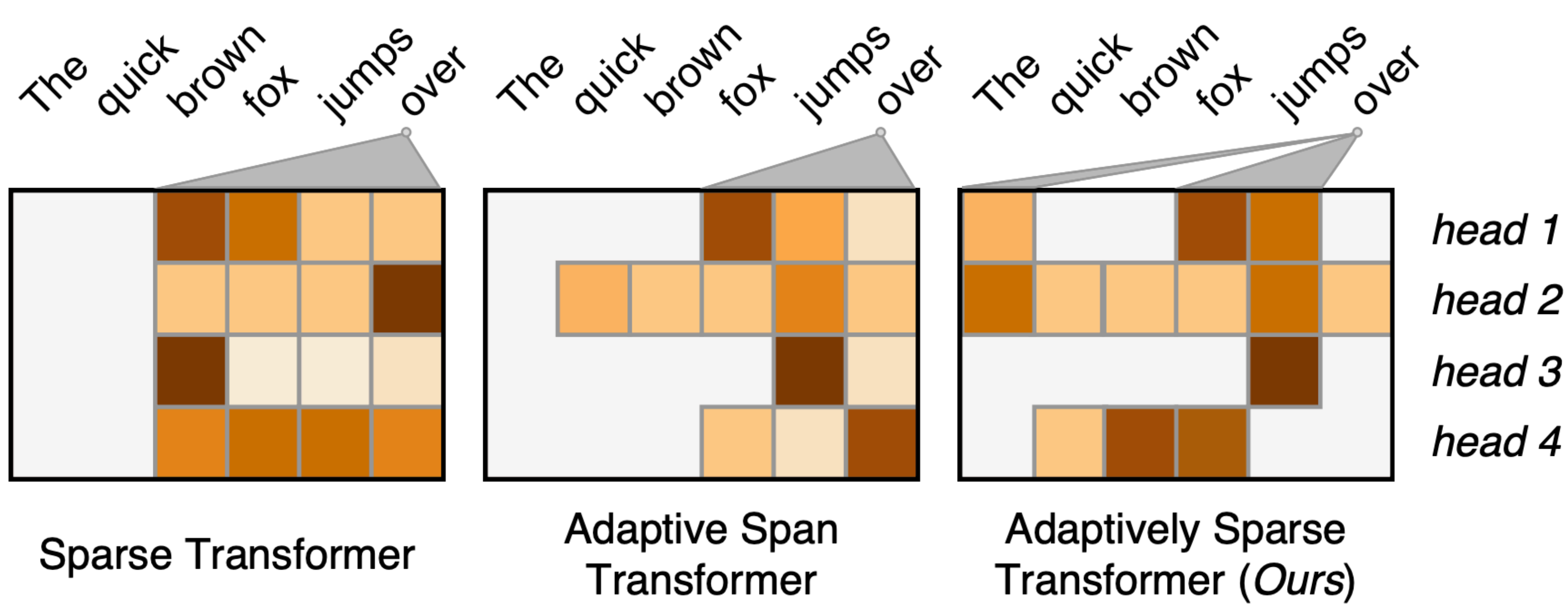


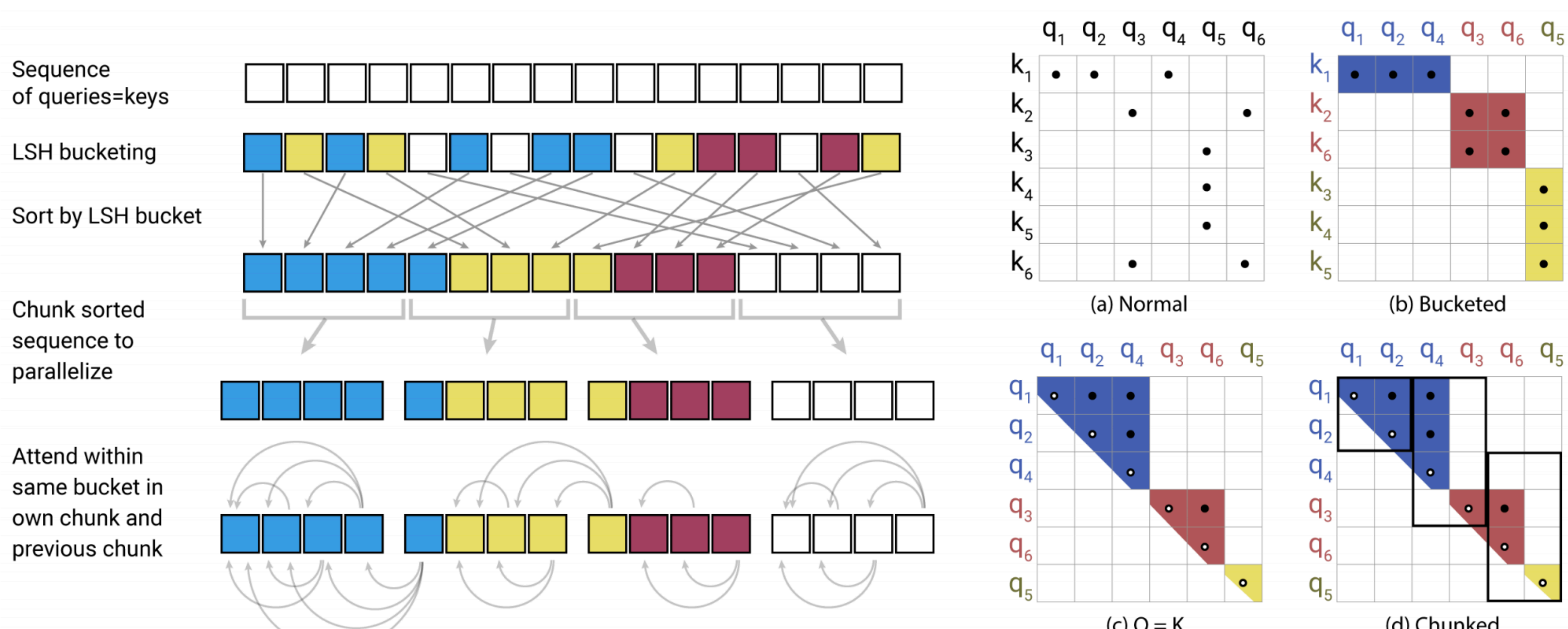
Figure 4: Adaptive spans (in log-scale) of every attention heads in a 12-layer model with span limit $S = 4096$. Few attention heads require long attention spans.

- 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 calculation of sparse attention 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

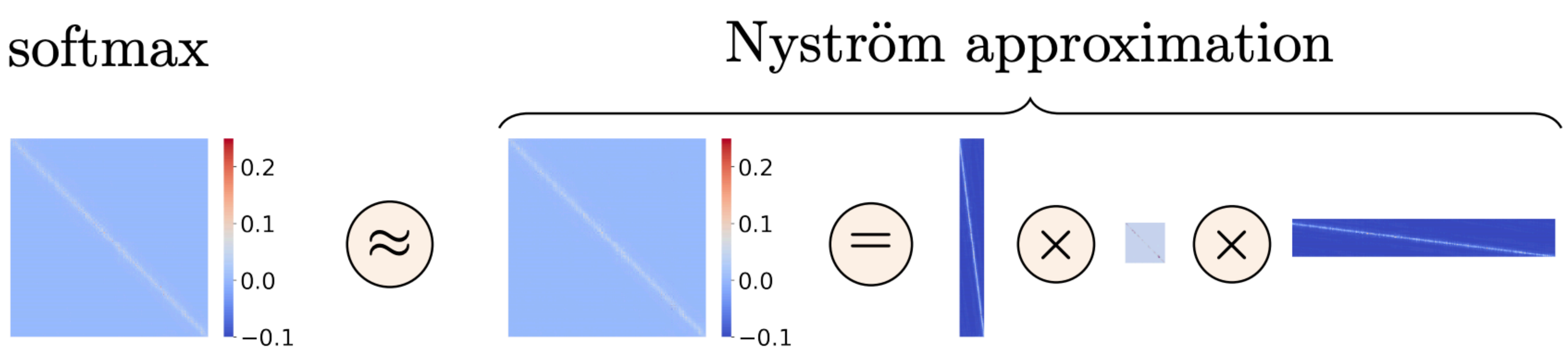


Low-rank Approximation

- Calculating the attention matrix is expensive, can it be predicted with a low-rank matrix?

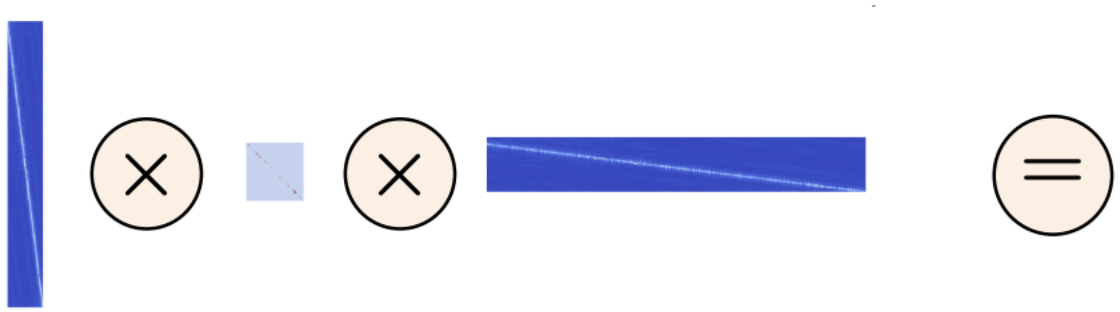
- **Linformer:** Add low-rank linear projections into model (Wang et al. 2020)

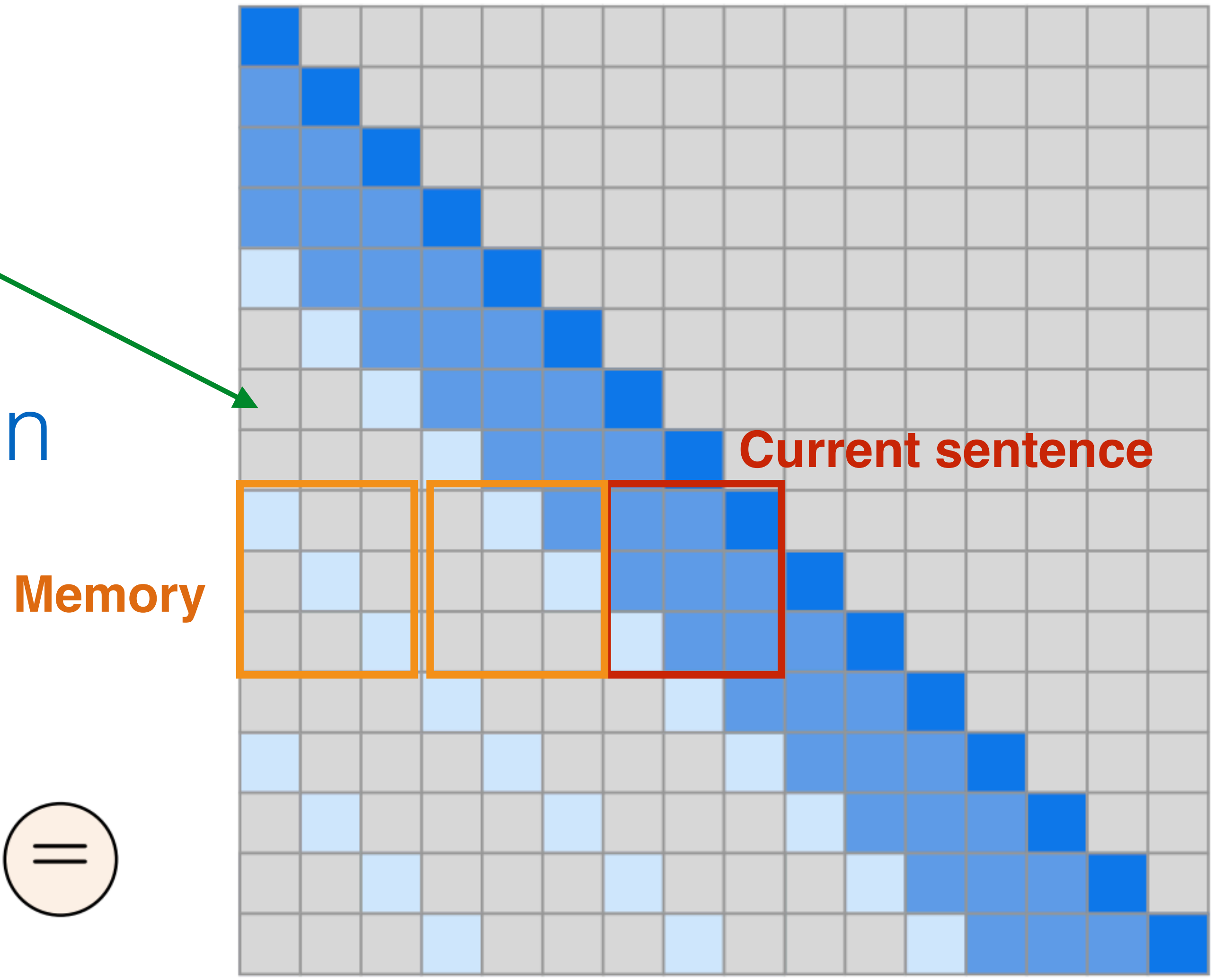
- **Nystromformer:** Approximate using the Nystrom method, sampling "landmark" points (Xiong et al. 2021)



Summary

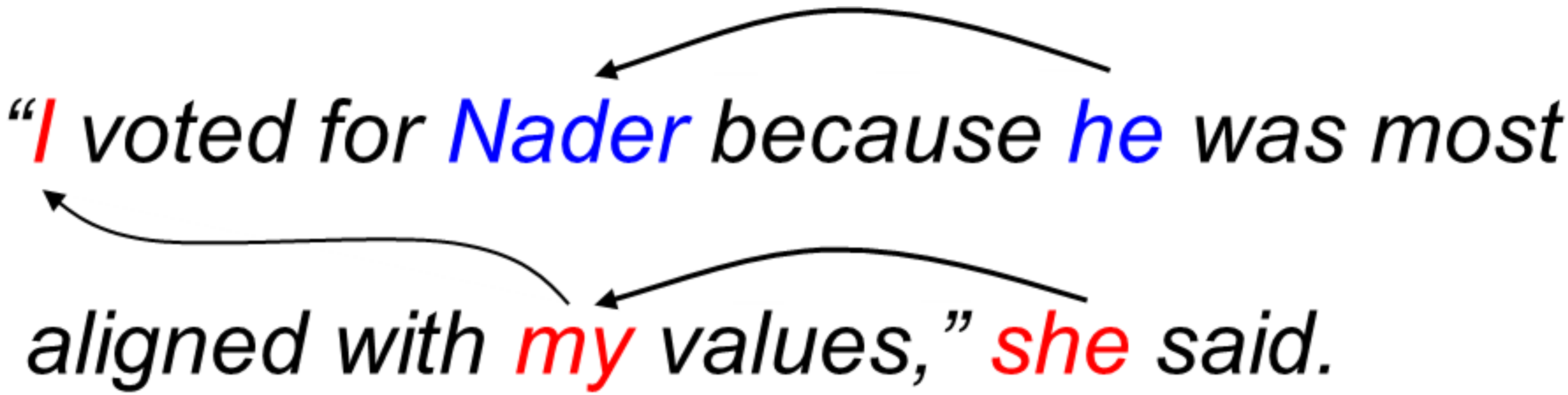
- The current bottleneck of Transformer-based model for long sequences is the **computation of attention matrix**
    - Attend to past memory
    - Sparse assumption
    - Low-rank approximation
    - …





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)



Entity Coreference

Image credit: Stanford NLP

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 **instances** (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

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.

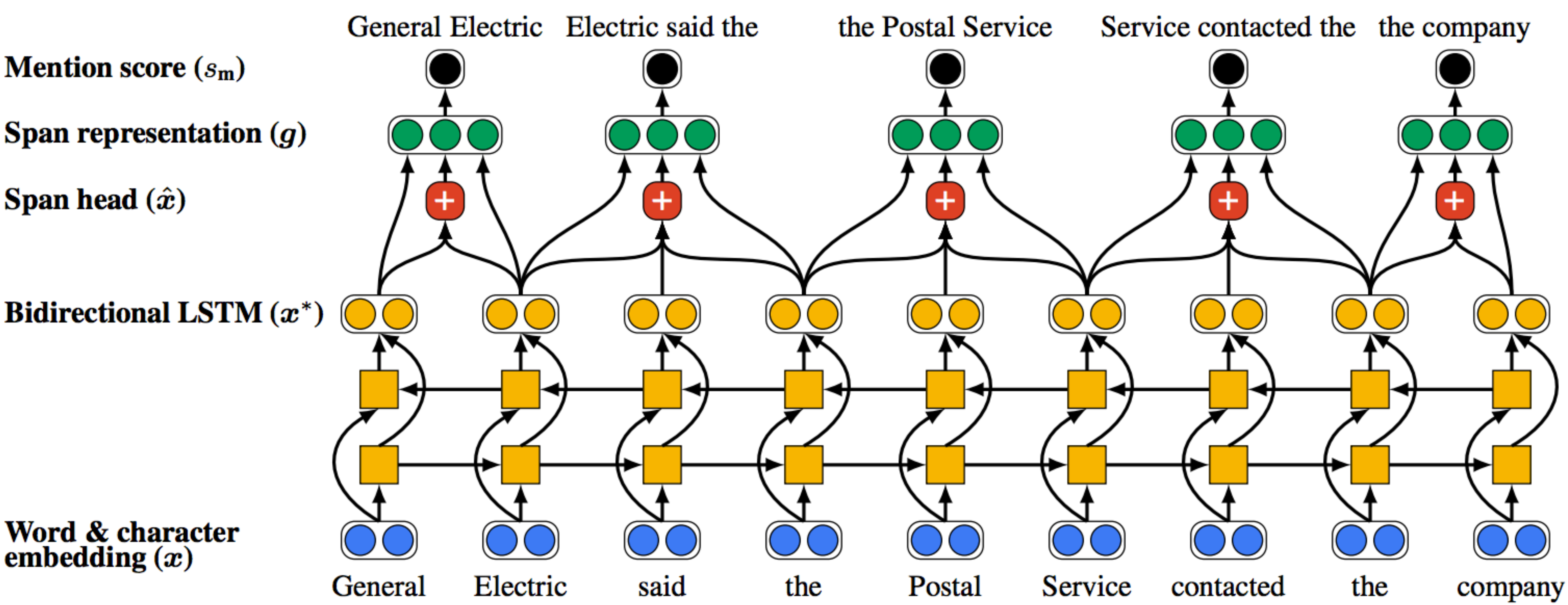
- **Jointly perform mention detection** and clustering.

End-to-End Neural Coreference

Lee et.al (2017)

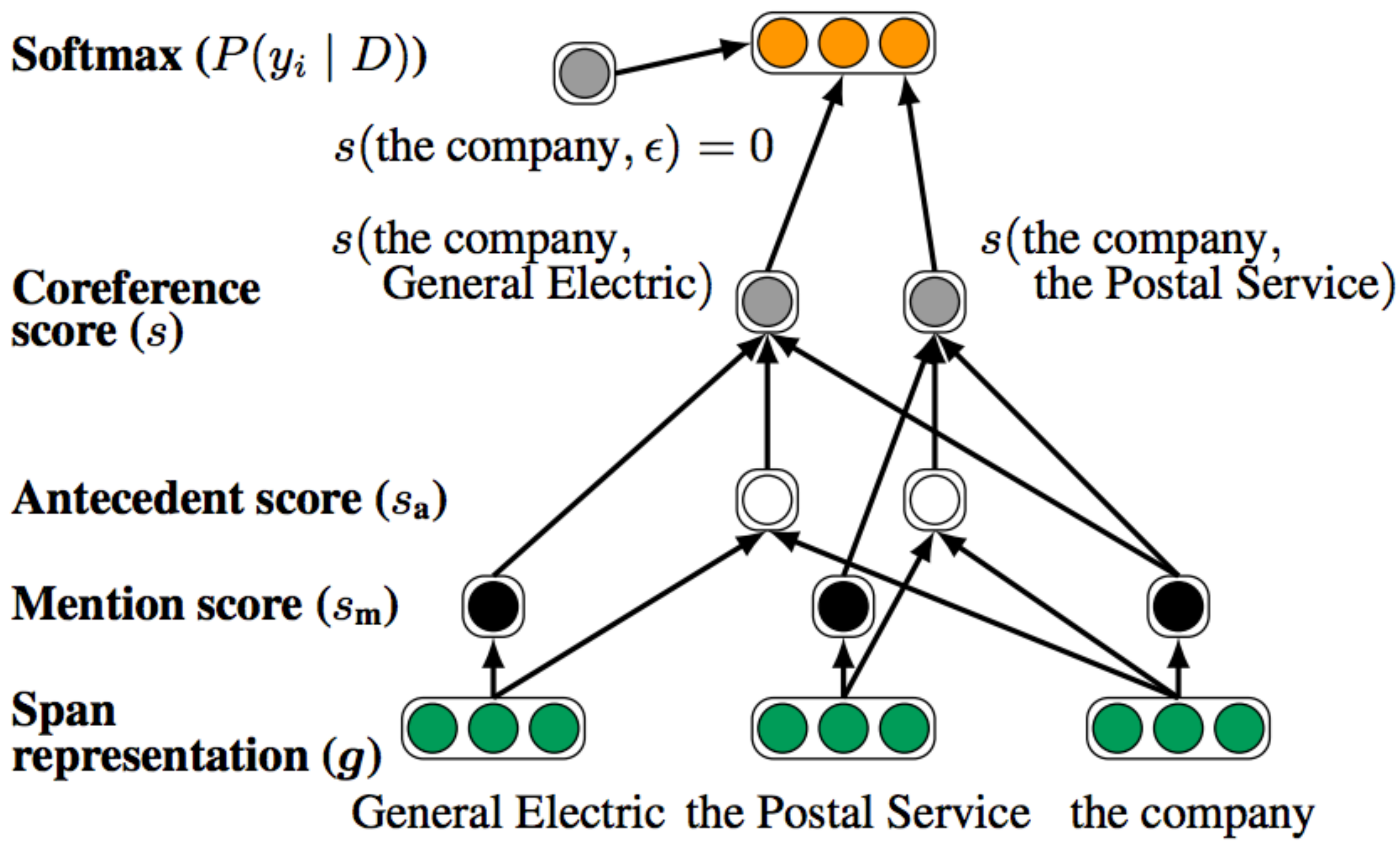
- 2 main contributions by this paper:
  - Learning all features of spans by a more typical neural network
  - Neural network allow errors to flow end-to-end. Jointly train with mention detection and coreference label classification
    - + This solves another type of error (span error), which is not previously handled.

End-to-End Neural Coreference (Span Model)

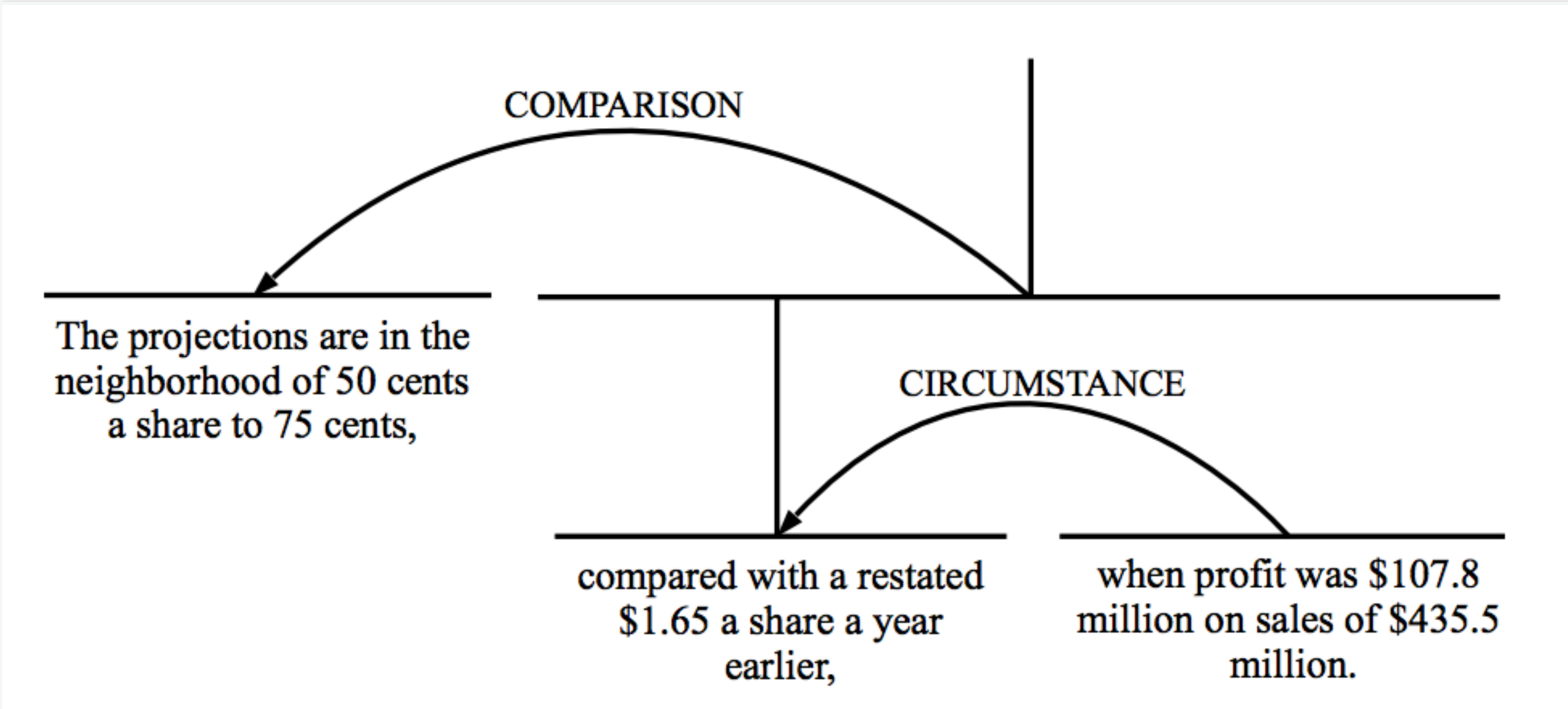


- Build mention representation from word representation (all possible spans)
- Head extracted by self-attention.

End-to-End Neural Coreference (Coreference Model)

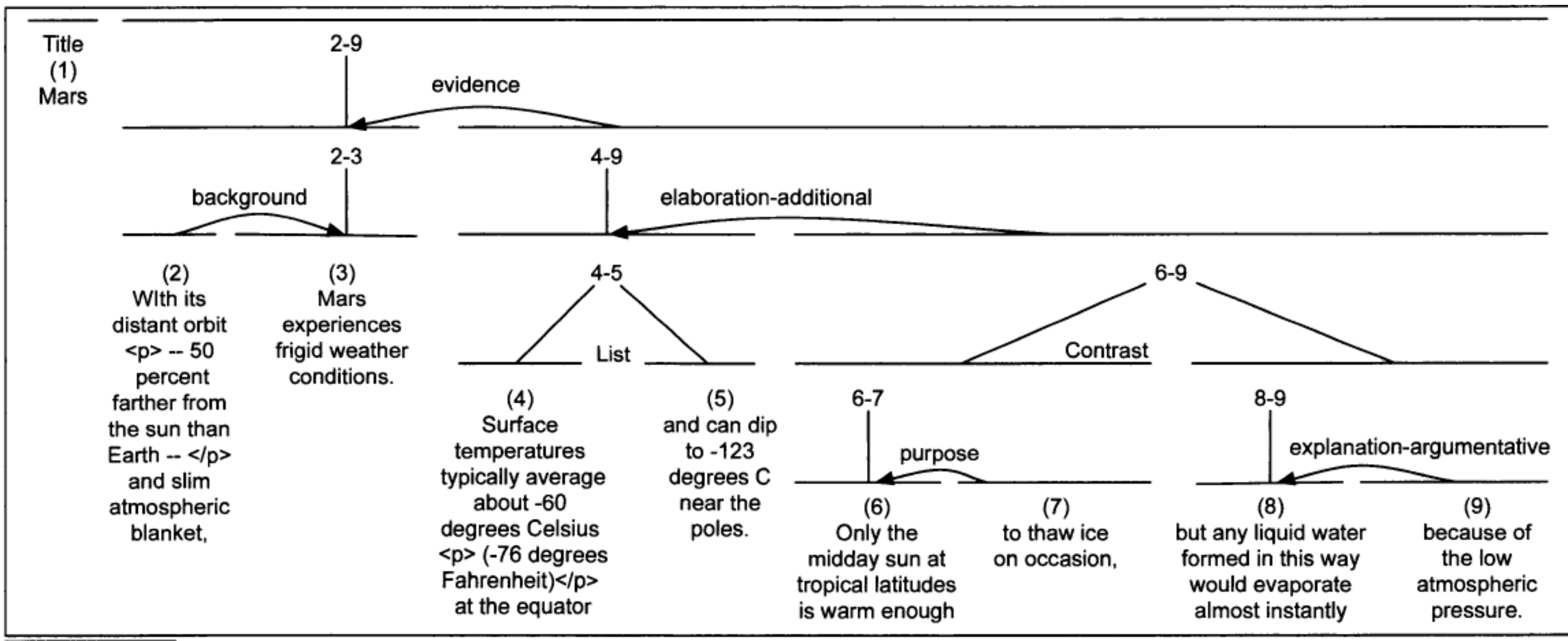


- Coreference model is similar to a mention ranking.
- Coreference score consist of multiple scores.
- Simple max-likelihood



Discourse Parsing

Document Problems: Discourse Parsing

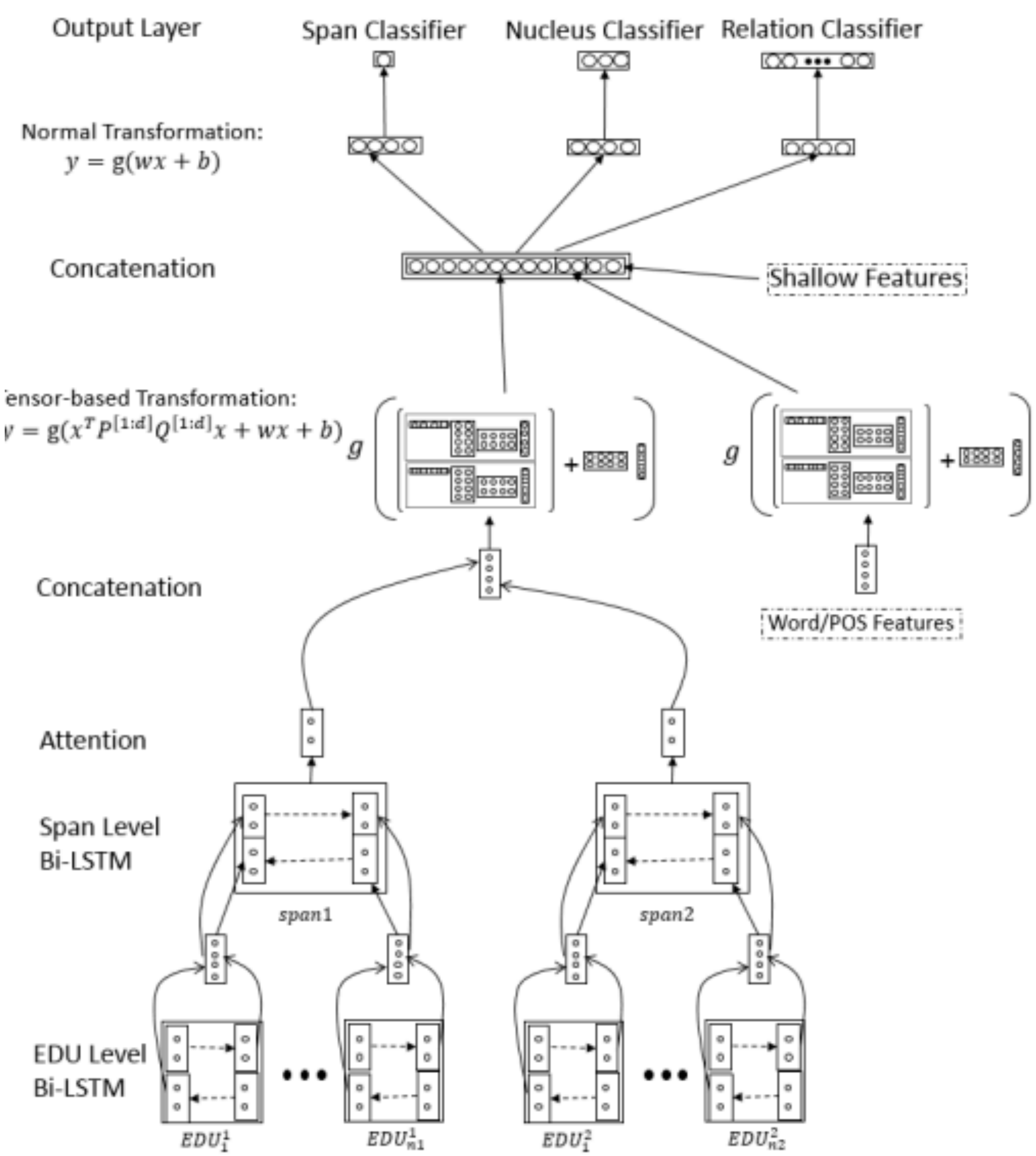


- Parse a piece of text into a relations between **elementary discourse units (EDUs)**.
- Researchers mainly used the Rhetorical Structure Theory (RST) formalism, which forms a tree of relations.

Example RST structures from Marcu (2000)

Discourse Parsing w/ Attention-based Hierarchical Neural Networks

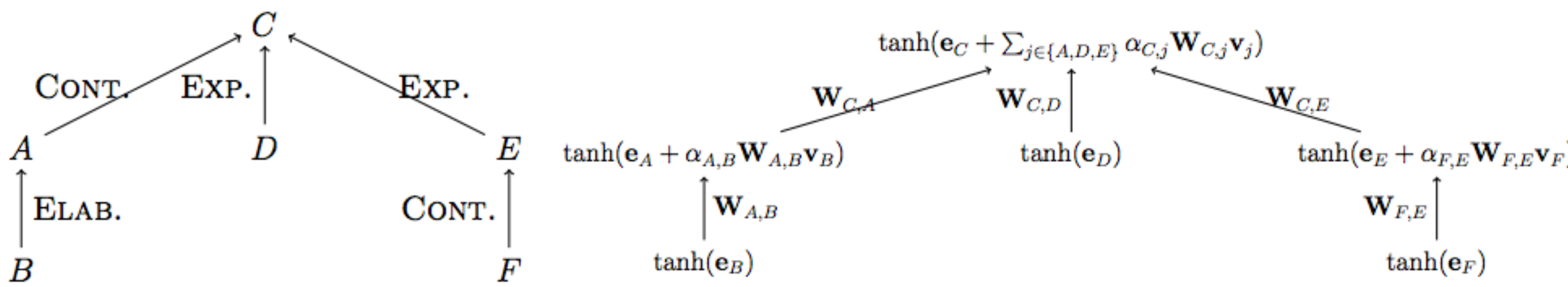
Li et.al (2016)



- Hierarchical bi-LSTM to learn composition scoring.
- Augmented with attention mechanism. (Span is long)
- 2 Bi-LSTMs: first used to capture the representation of a EDU, then combine EDU representation into larger representation
- CKY Parsing

Uses of Discourse Structure in Neural Models

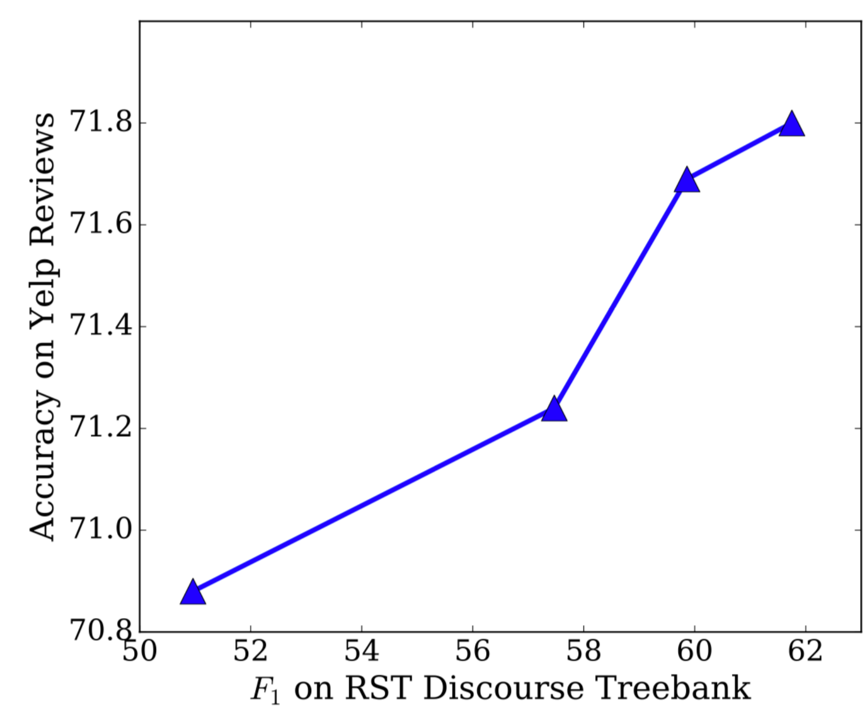
- Discourse-structured classification with neural models (Ji and Smith 2017)



(a) dependency structure

(b) recursive neural network structure

- Good results, and more interestingly, discourse parsing accuracy very important!



Questions?