CS769 Advanced NLP

Modeling Long Sequences

Junjie Hu

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Slides adapted from Zhengzhong, Graham

https://junjiehu.github.io/cs769-fall25/

Goal for Today

1. Document-level Neural Language Modeling

    - RNN-based Models
    - Transformer-based Models

2. Other Document-level Tasks

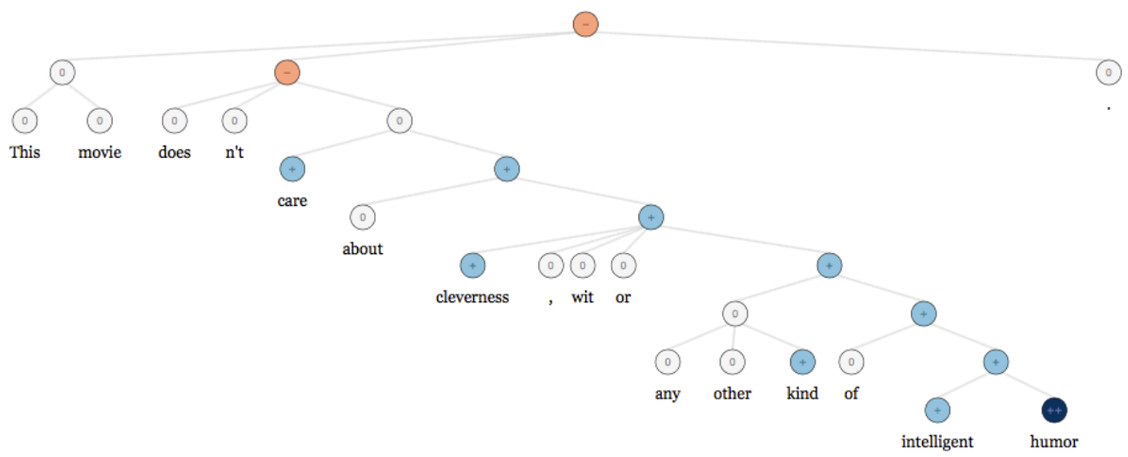
    - Entity Coreference
    - Discourse Parsing

Some NLP Tasks we've Handled

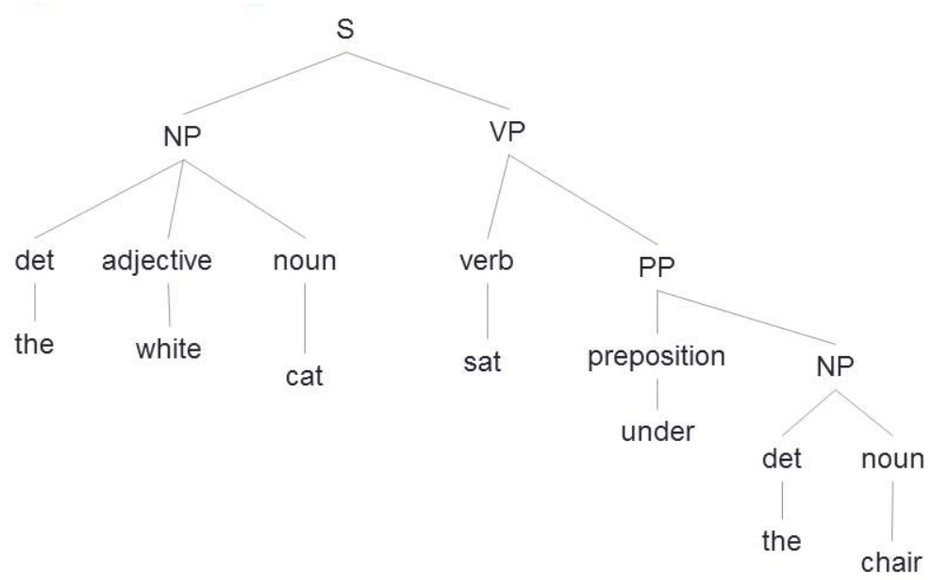
Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

$P(w_{i+1}= \text{of} \mid w_i=\text{tired}) = 1$
$P(w_{i+1}= \text{of} \mid w_i=\text{use}) = 1$
$P(w_{i+1}= \text{sister} \mid w_i=\text{her}) = 1$
$P(w_{i+1}= \text{beginning} \mid w_i=\text{was}) = 1/2$
$P(w_{i+1}= \text{bank} \mid w_i=\text{the}) = 1/3$
$P(w_{i+1}= \text{book} \mid w_i=\text{the}) = 1/3$
$P(w_{i+1}= \text{use} \mid w_i=\text{the}) = 1/3$

Language Models



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

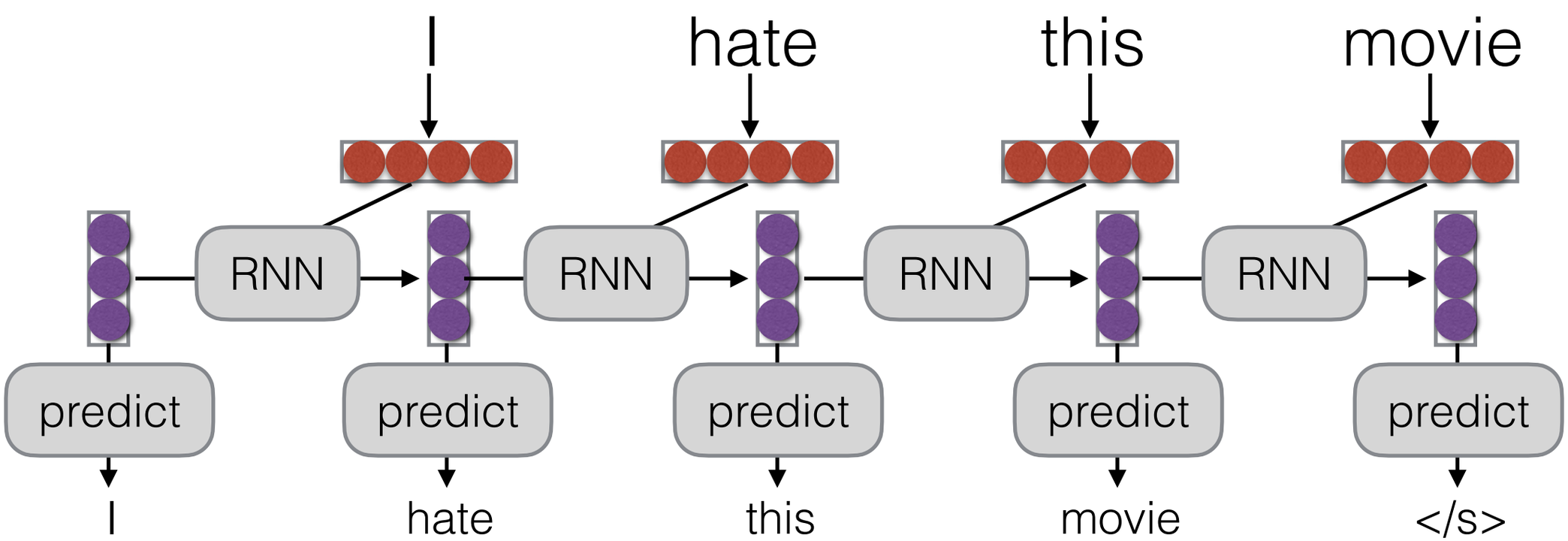
Long-Context Language Modeling

- We want to predict the probability of words in an long document

- Obviously sentences in a document don't exist in a vacuum! We want to take advantage of this fact.

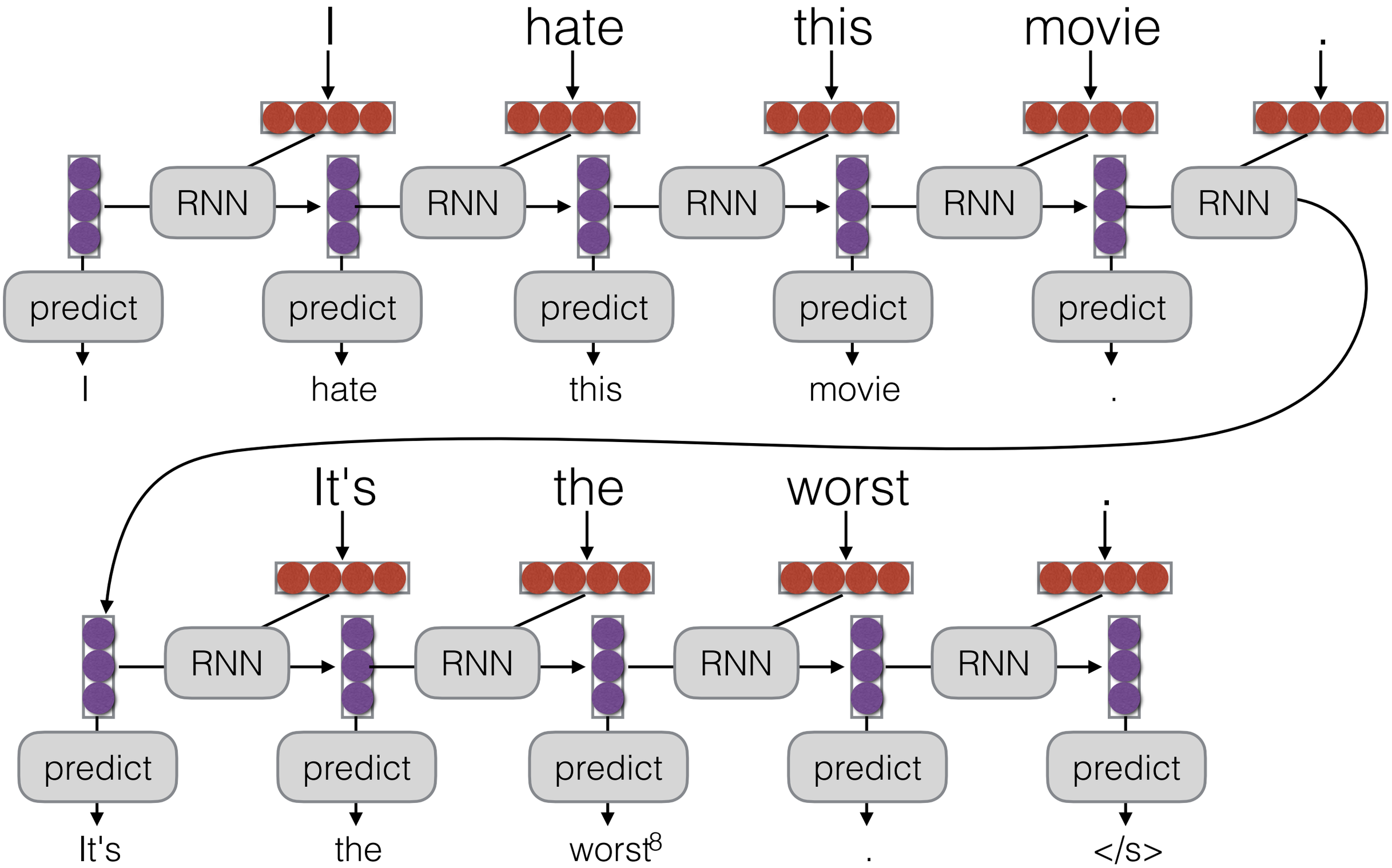
Remember: Modeling using Recurrent Networks

- Model passing previous information in hidden state



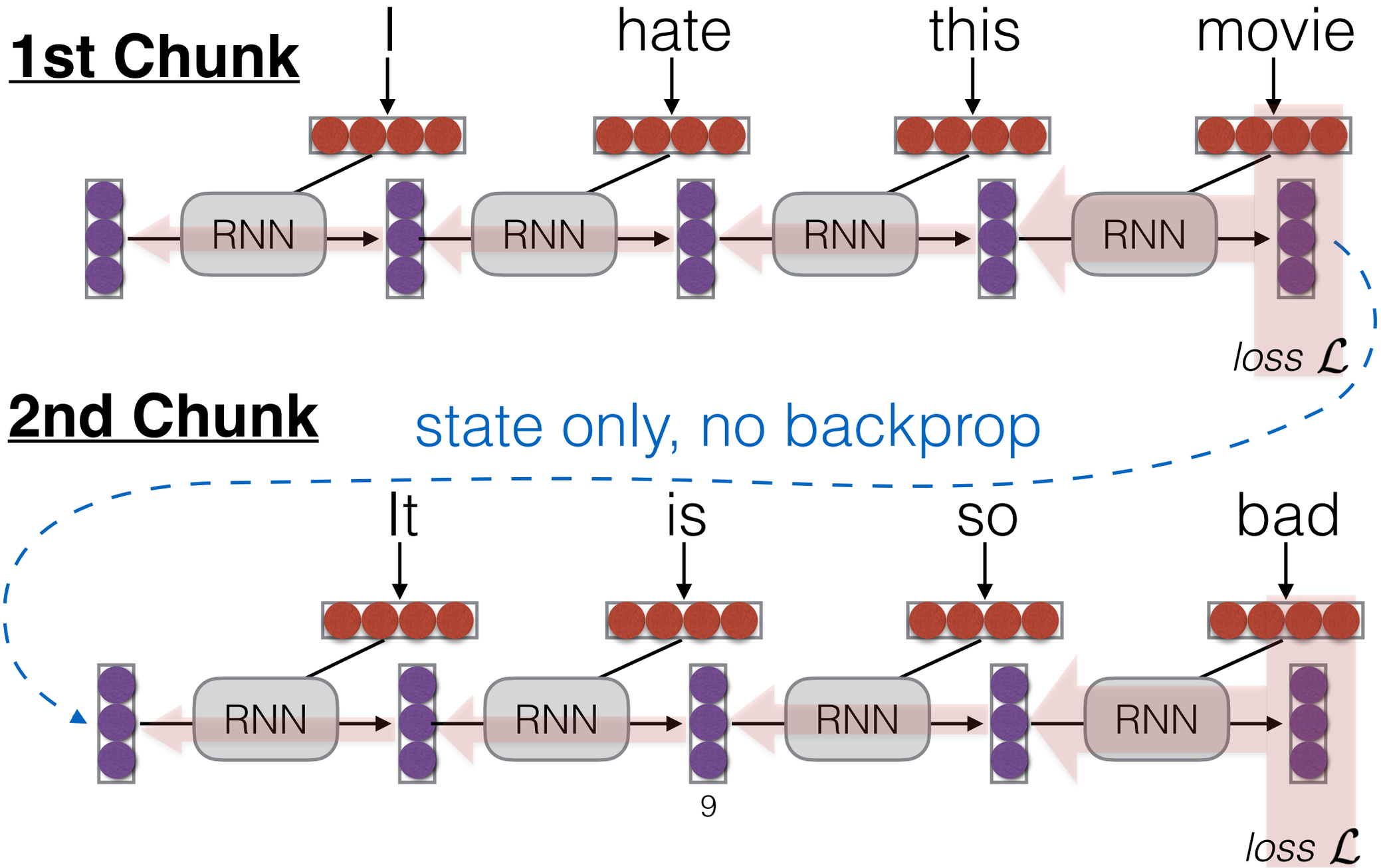
Simple: Infinitely Pass State by RNN LM

(Mikolov et al. 2011)



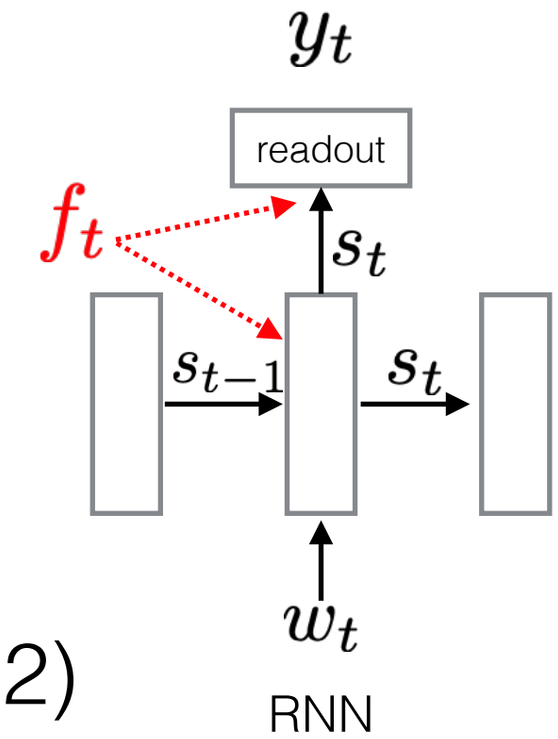
Truncated Backpropagation Through Time (TBPTT) (Elman 1990, Boden 2001)

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



Separate Encoding for Coarse-grained Document Context

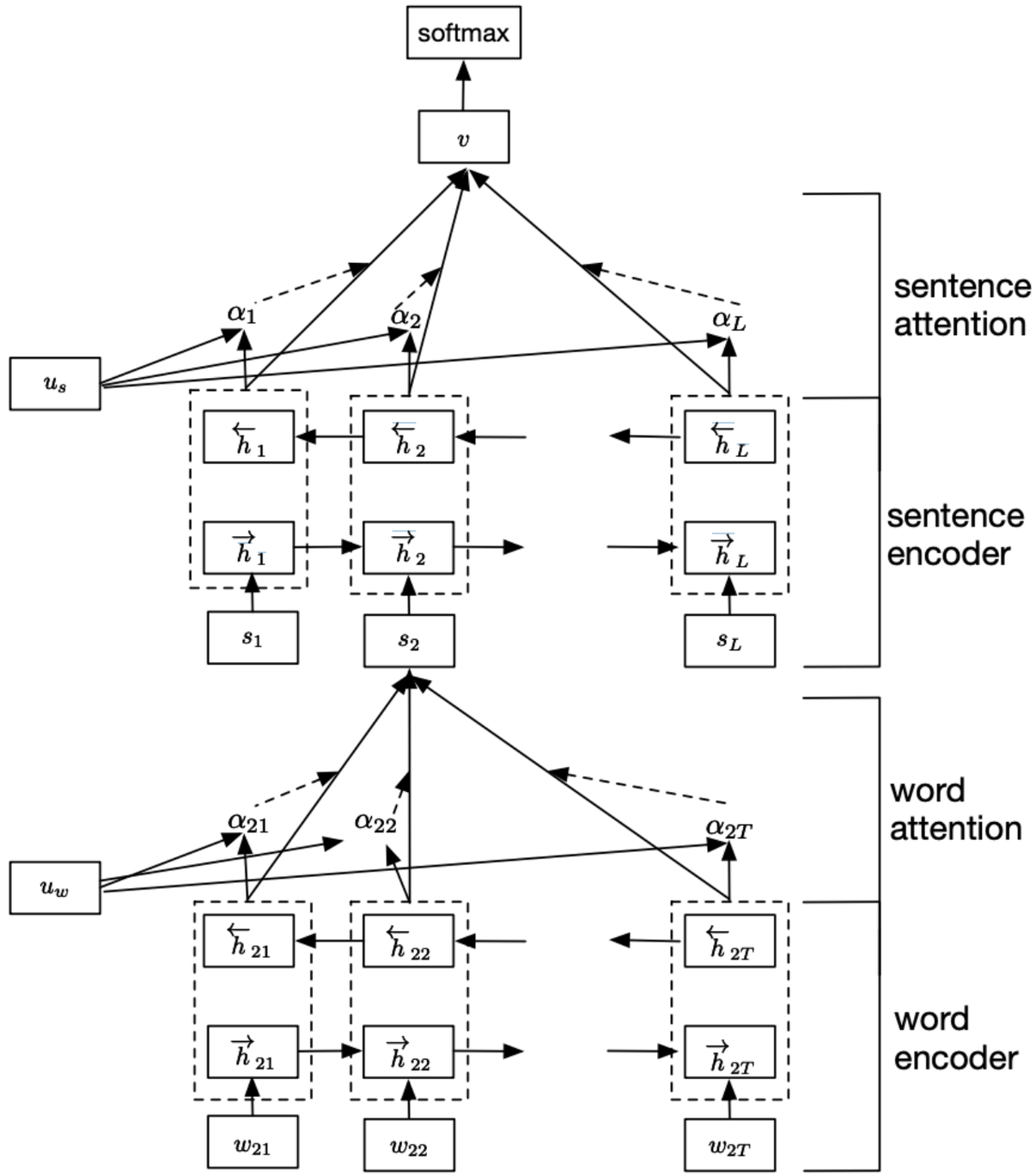
- Explicitly add the external global features f_t as input to
  1. each RNN cell
  2. The final readout linear layer
- What global context?
  - Use topic modeling (Mikolov & Zweig 2012)
  - Use bag-of-words of previous sentence(s), optionally with attention (Wang and Cho 2016)
  - Use last state of previous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



Hierarchical Attention Network

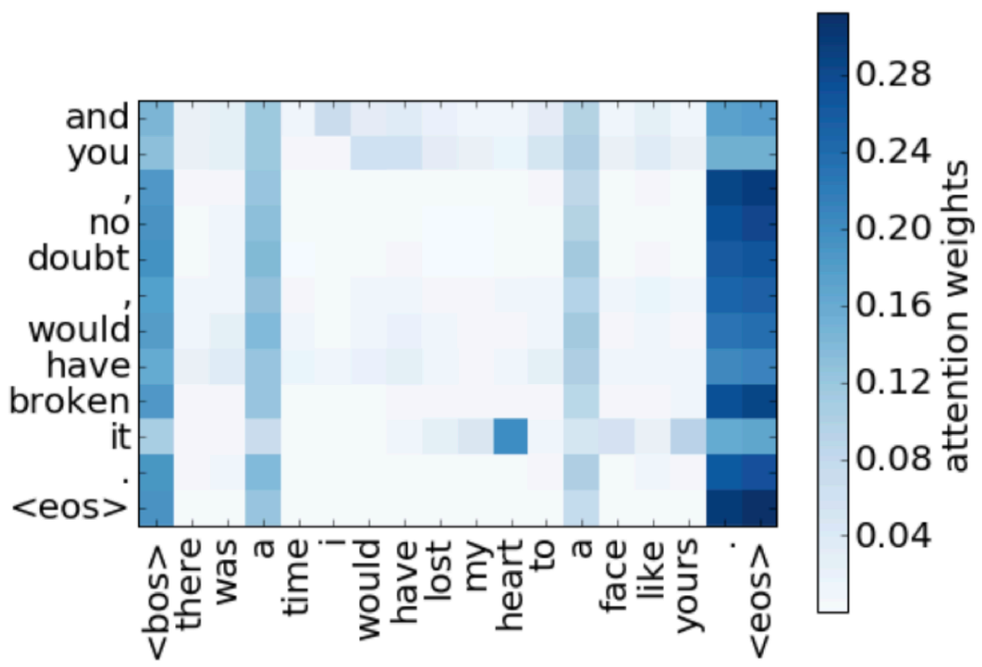
(Yang et al. 2018)

- 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

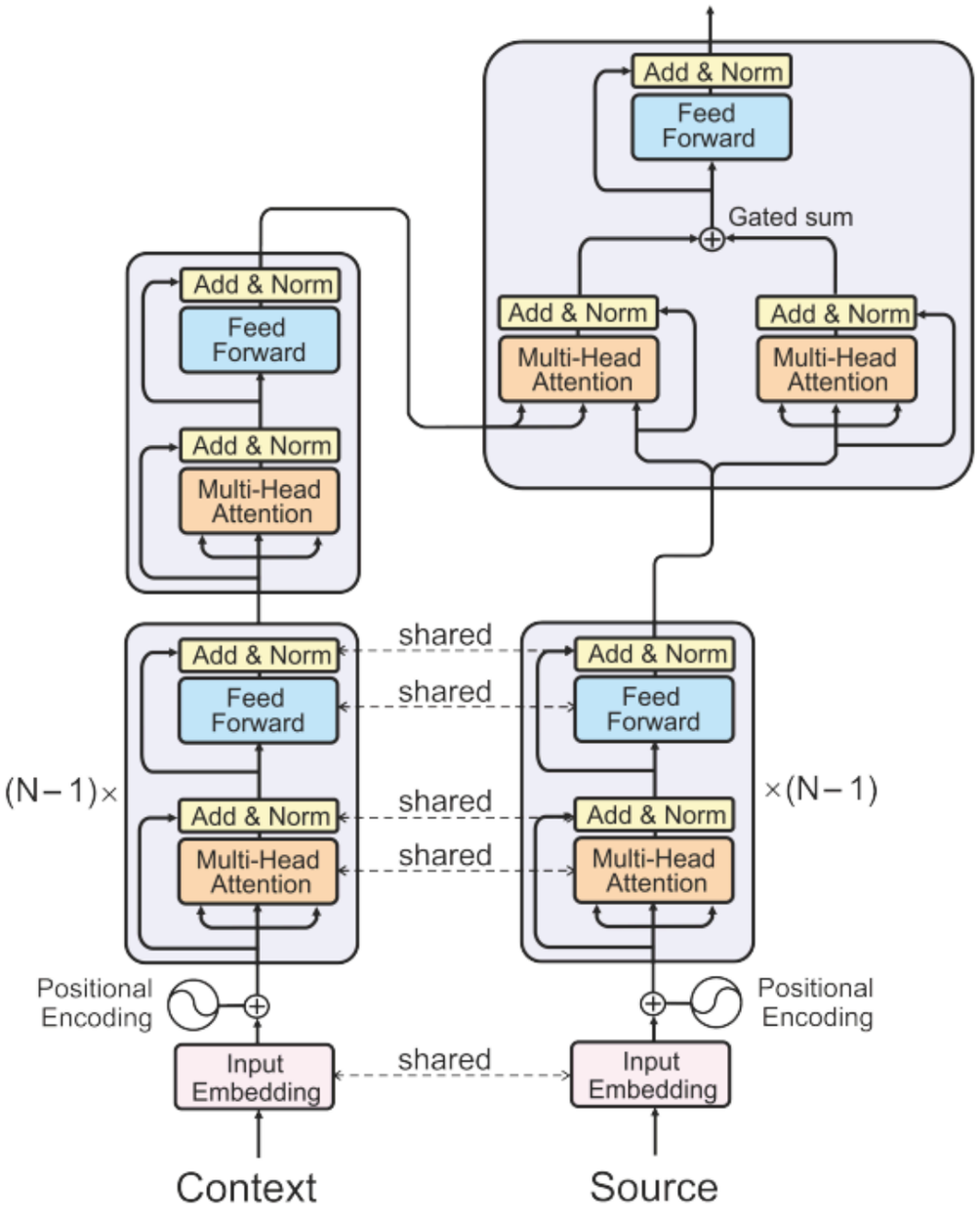
- Simply self-attend to all words in the document
  - + Can simply use document-level context
  - + Can learn interesting phenomena (e.g. co-reference)



- - 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** $O(l^2)$

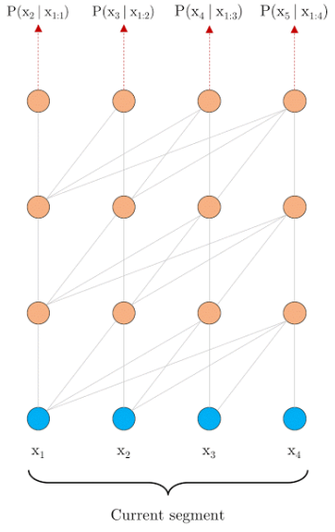


Transformer-XL: Truncated BPTT+Transformer

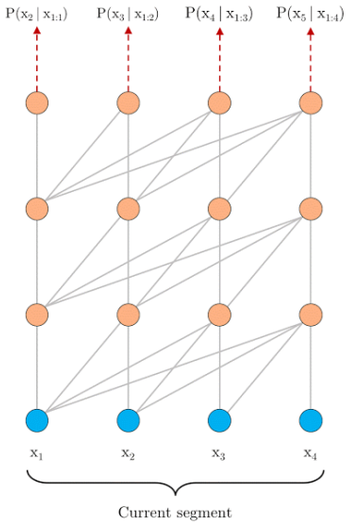
(Dai et al. 2019)

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

Standard Transformer



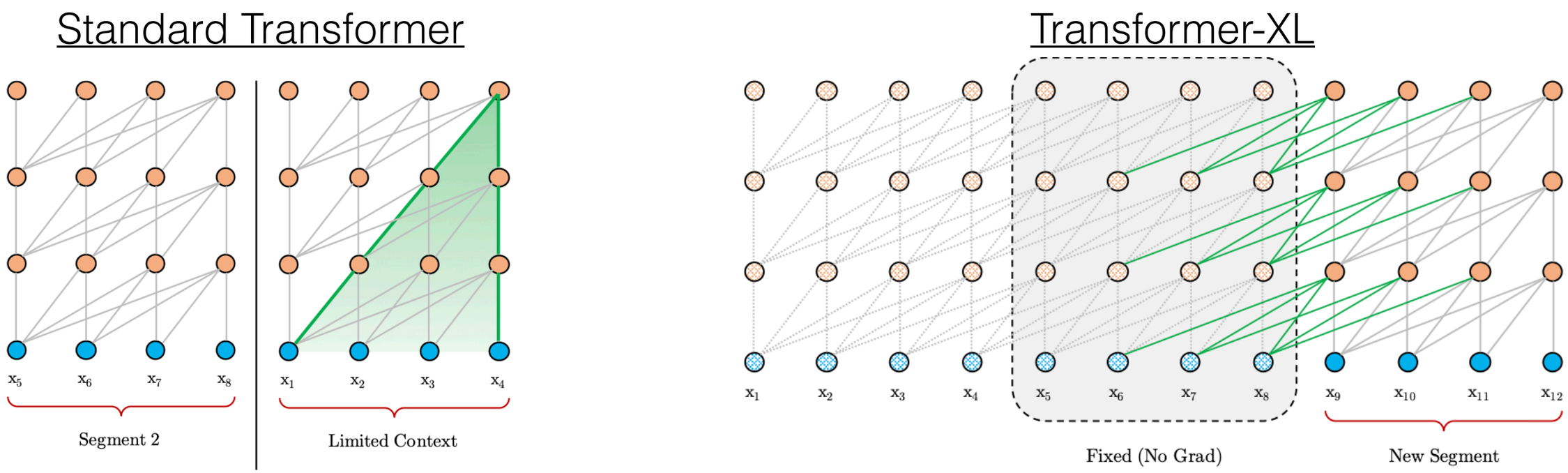
Transformer-XL



Transformer-XL: Truncated BPTT+Transformer

(Dai et al. 2019)

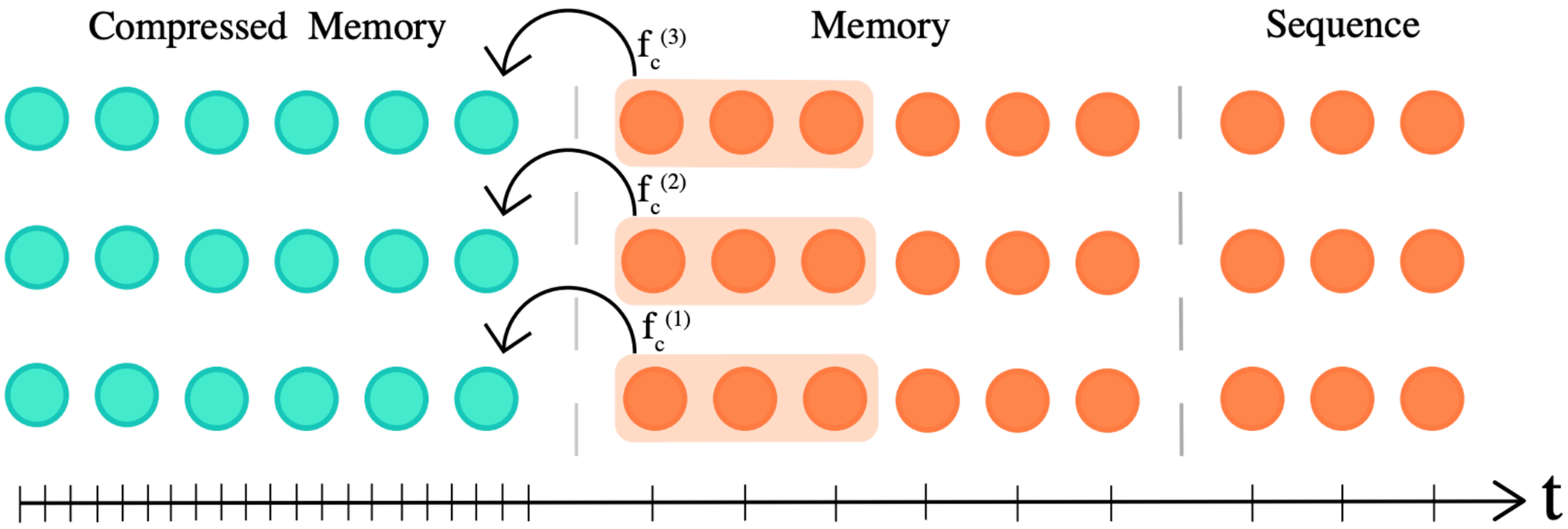
- Like truncated backprop through time for RNNs; can use previous states, but not backprop into them



- 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

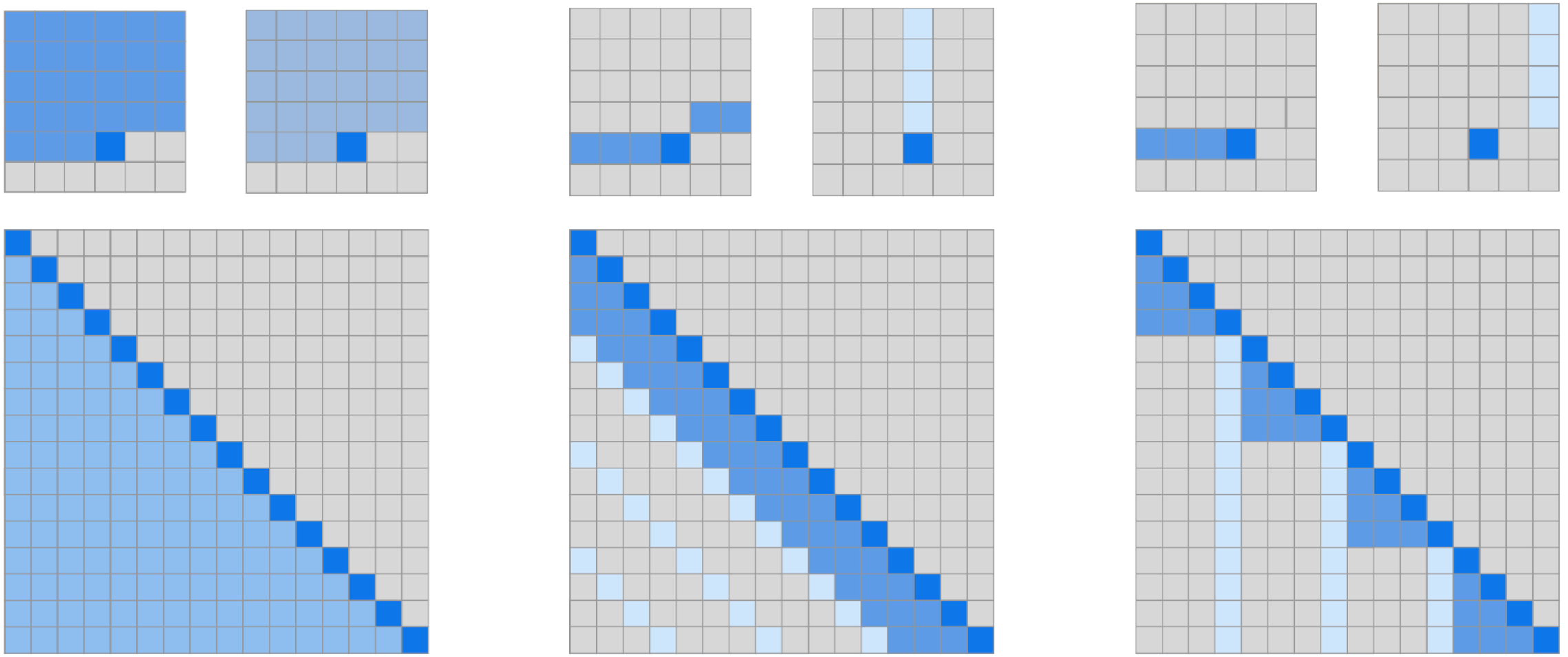
- Extension over Transformer-XL that uses the last chunk as the memory
- Add a "strided" compression step over previous states (Rae & Potapenko et al. 2019)



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)

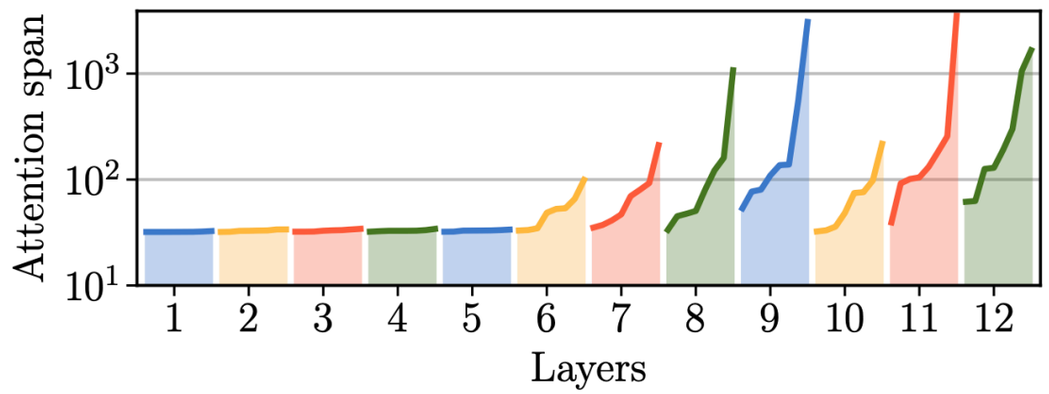
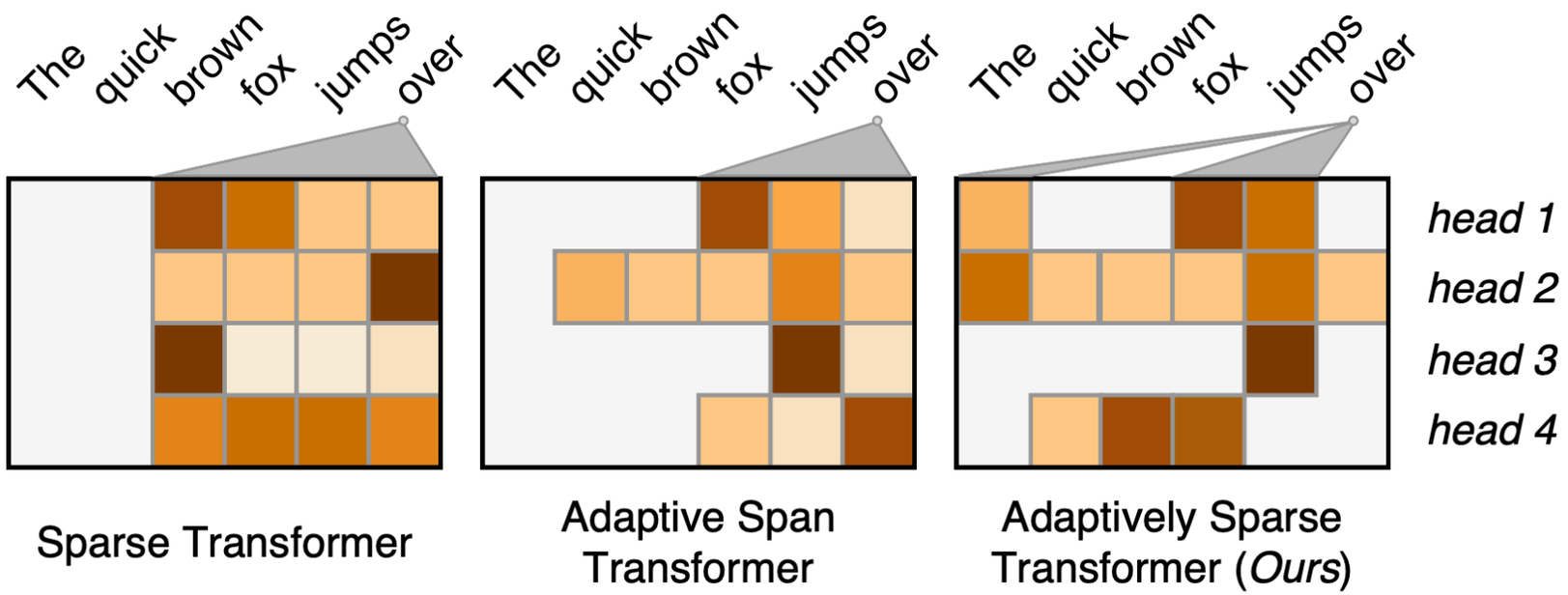


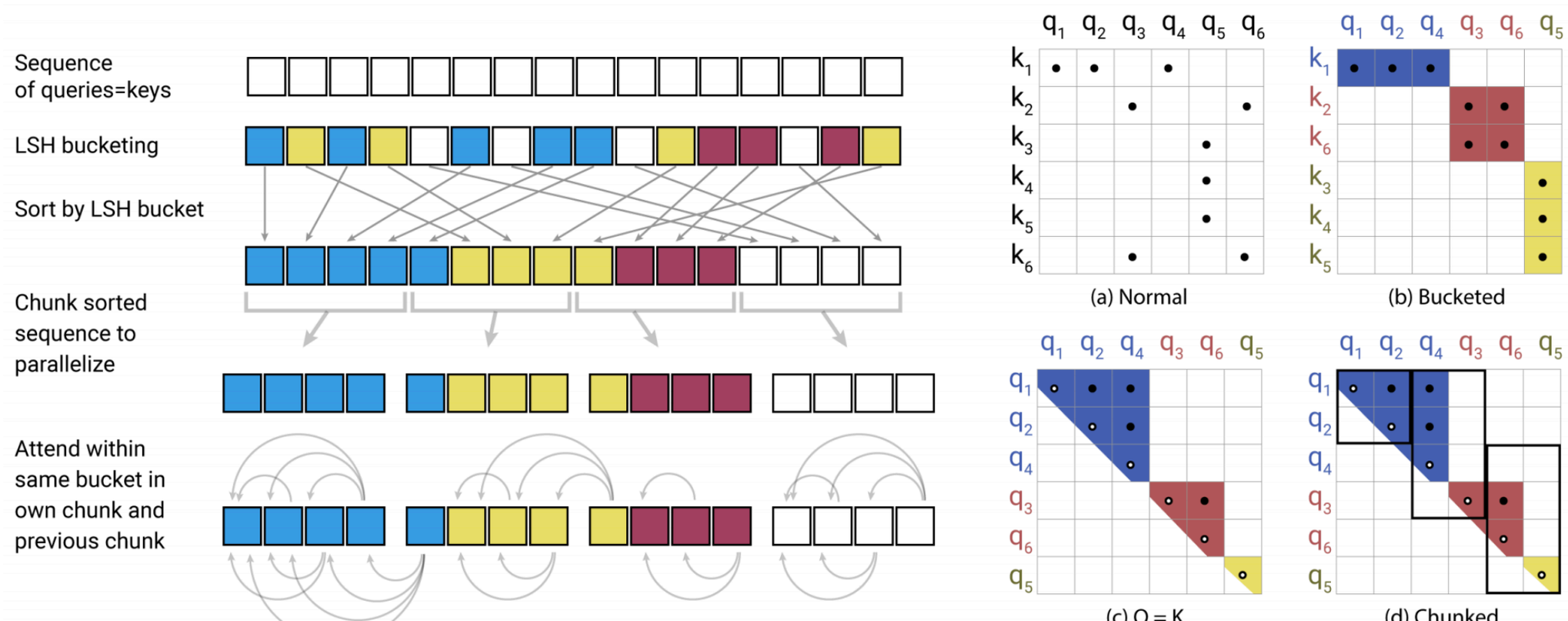
Figure 4: Adaptive spans (in log-scale) of every attention heads in a 12-layer model with span limit $S = 4096$. Few attention heads require long attention spans.

- 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

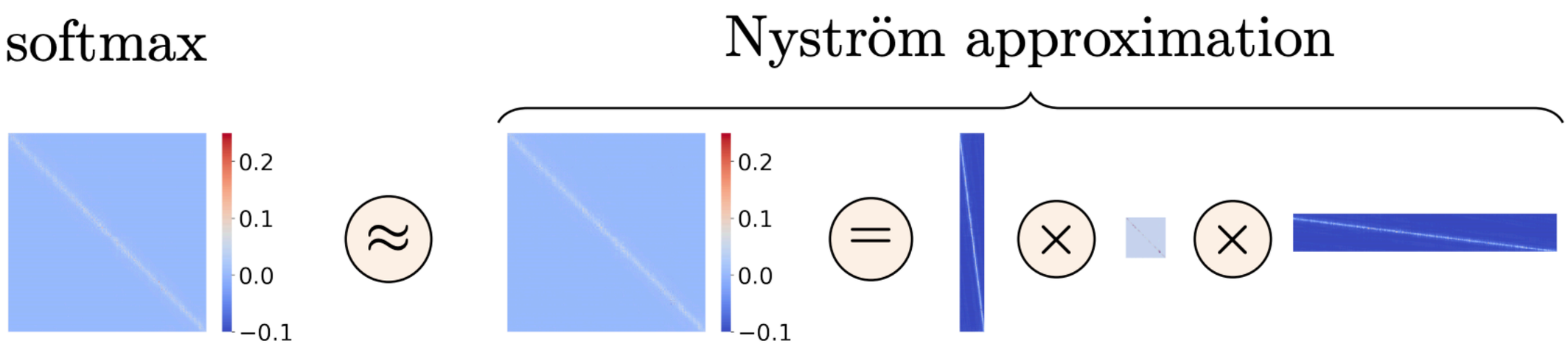


Low-rank Approximation

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

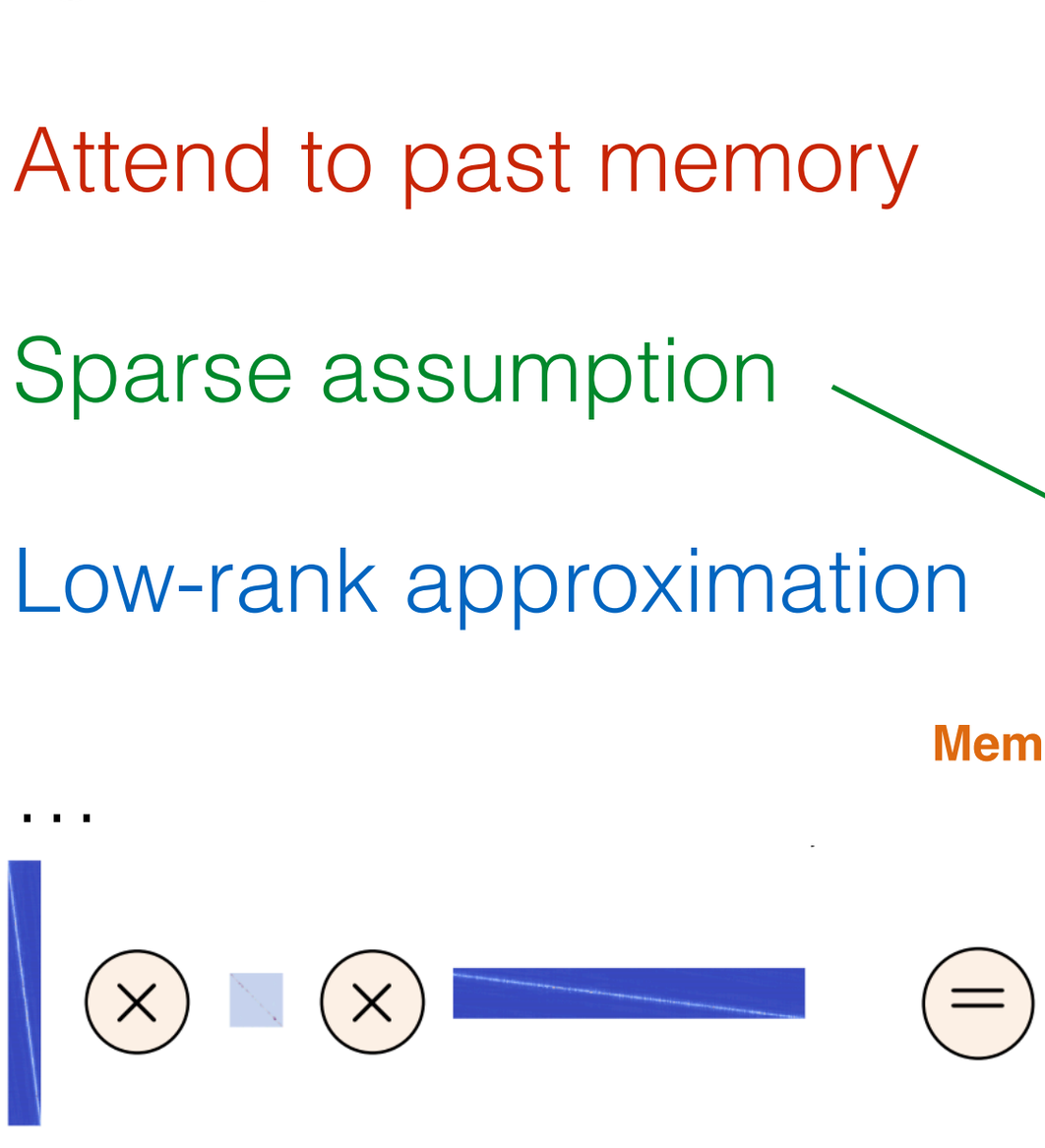
- **Linformer:** Add low-rank linear projections into model (Wang et al. 2020)

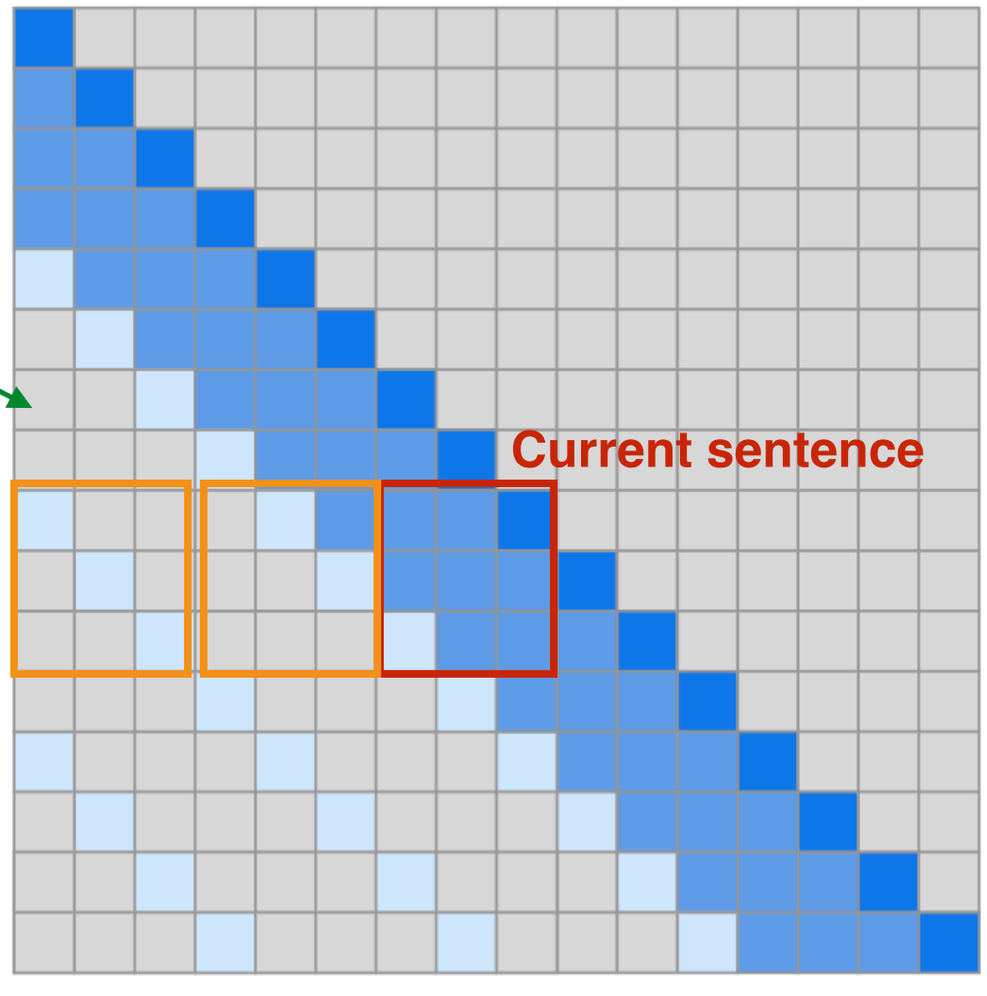
- **Nystromformer:** Approximate using the Nystrom method, sampling "landmark" points (Xiong et al. 2021)



Summary of Training-time Methods

- 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
  - …





Long-Context Language Modeling **at Inference**

- During pre-training, large language models (LLMs) are often using **full attention** **and** **a length limit** (i.e., without any modification to their attentions).

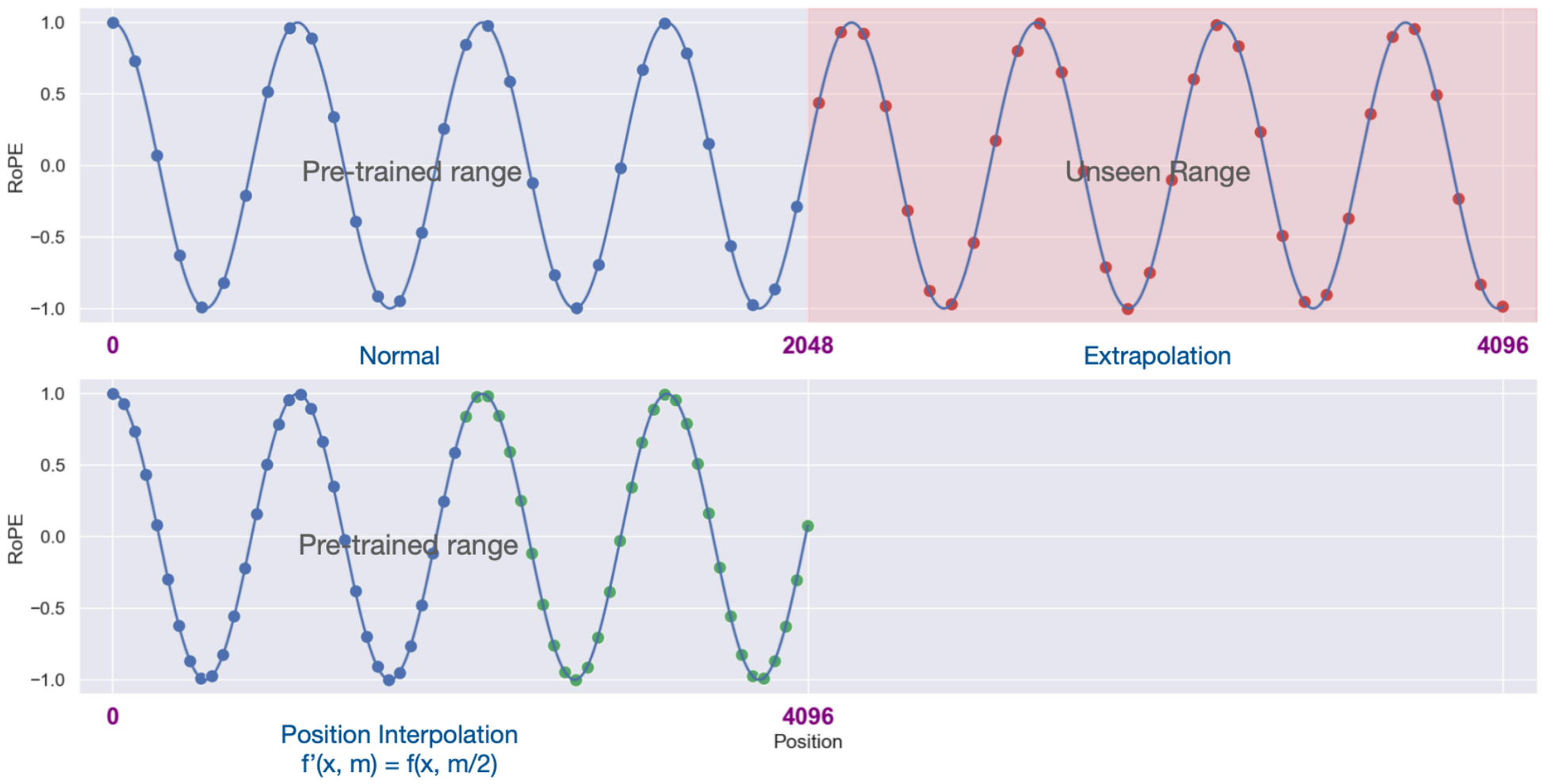
- What about extending the context length limit of a **trained LLM at inference**?

  - **Position Interpolation**: estimate the position embedding out of the maximum length limit

  - **KV Cache**: cache the important key-value pairs from the attention in the memory

Length Extension at Inference

- Position Interpolation: What about extending the context length of a **trained LLM at inference**?



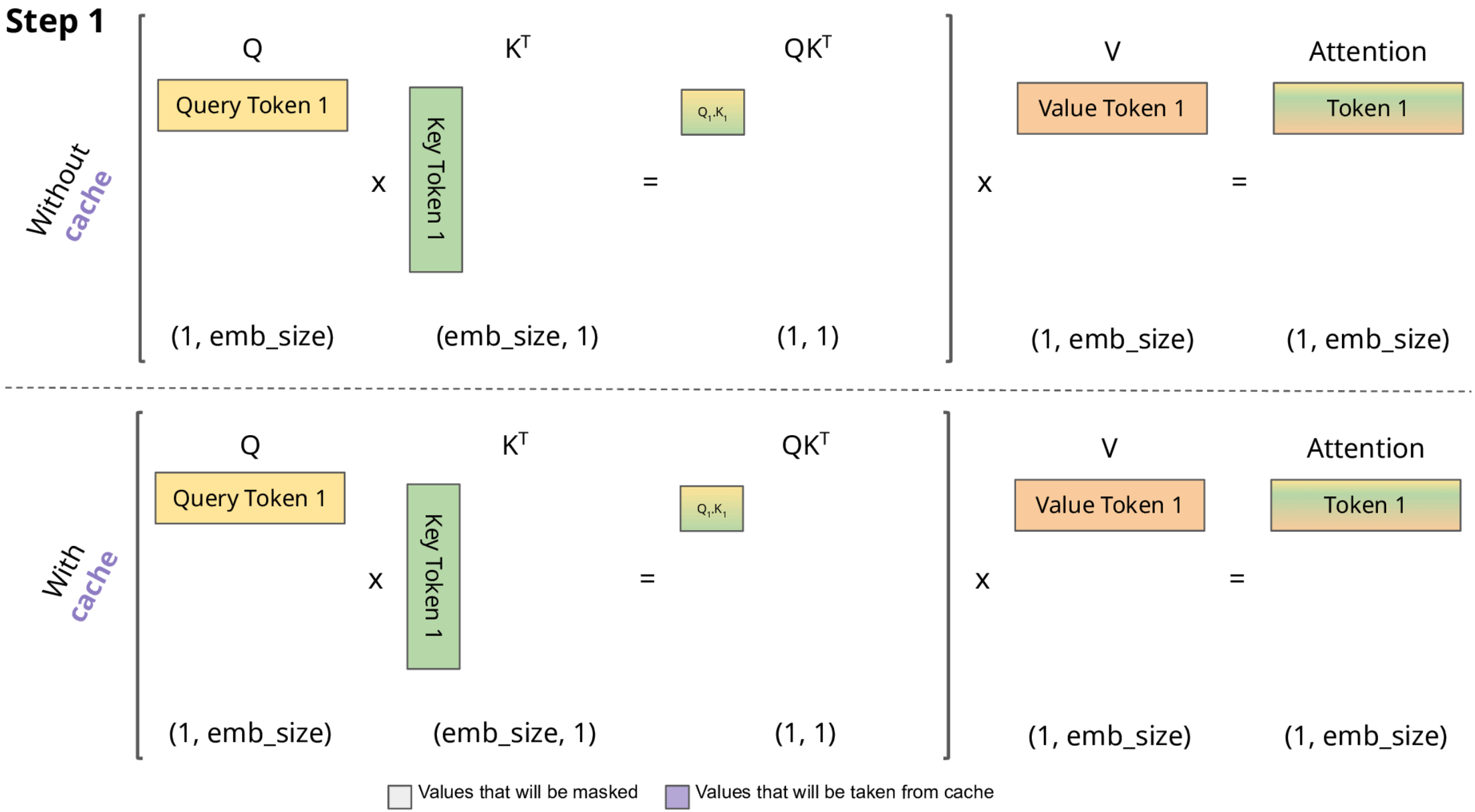
$$\mathbf{f}'(\mathbf{x}, m) = \mathbf{f}\left(\mathbf{x}, \frac{mL}{L'}\right).$$

f() is the function that takes in a word embedding x, and a position index, and returns an embedding with positional information

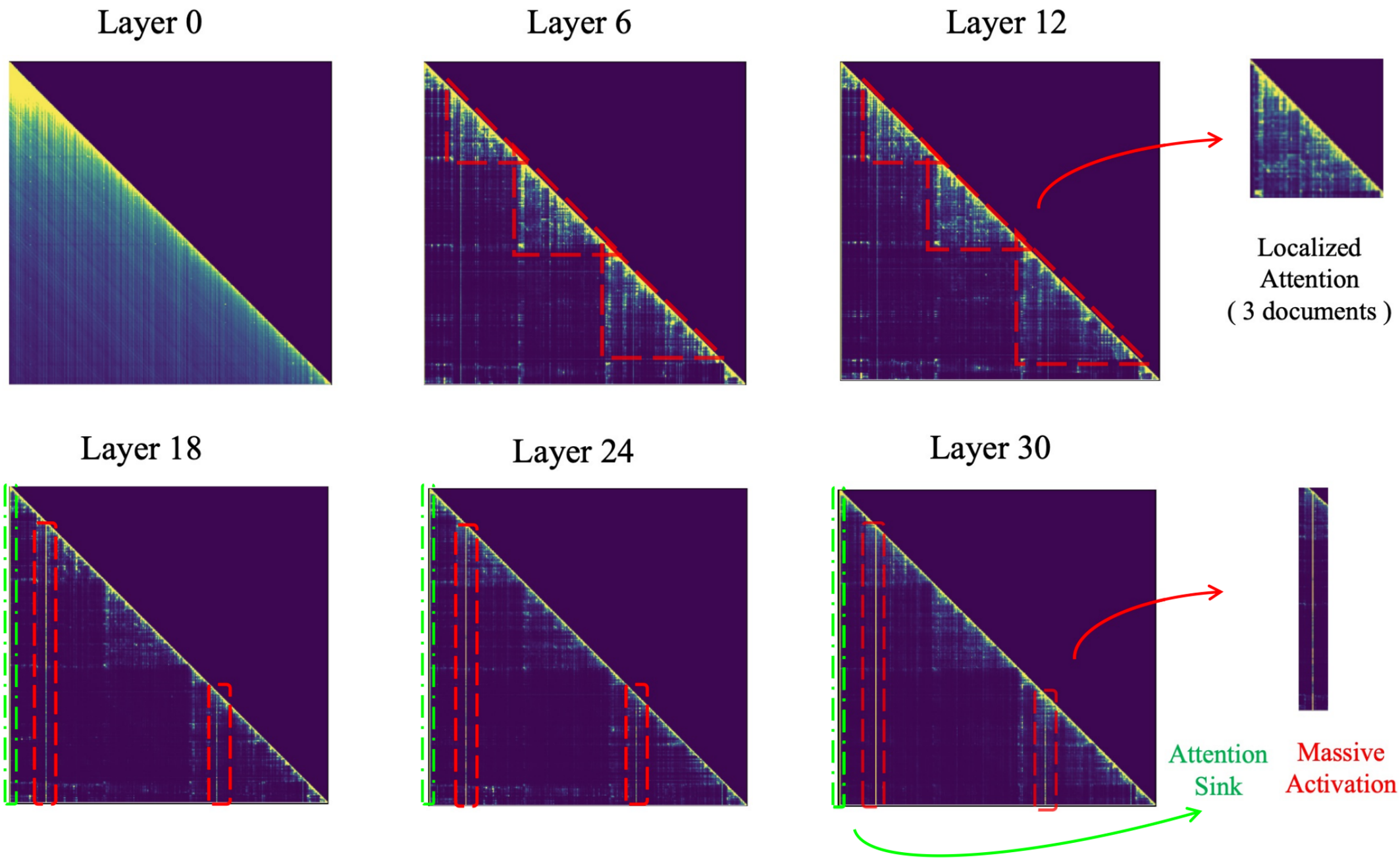
Chen et al. 2023 Extending Context Window of Large Language Models via Positional Interpolation

KV Cache

- Cache the computed key-value pairs from the long context



Attentions are not evenly distributed over layers



Cai, et al. 2024. PyramidKV: Dynamic KV Cache Compression based on Pyramidal Information Funneling

Existing KV Cache Methods

- Different strategies to cache KV pairs over transformer layers.

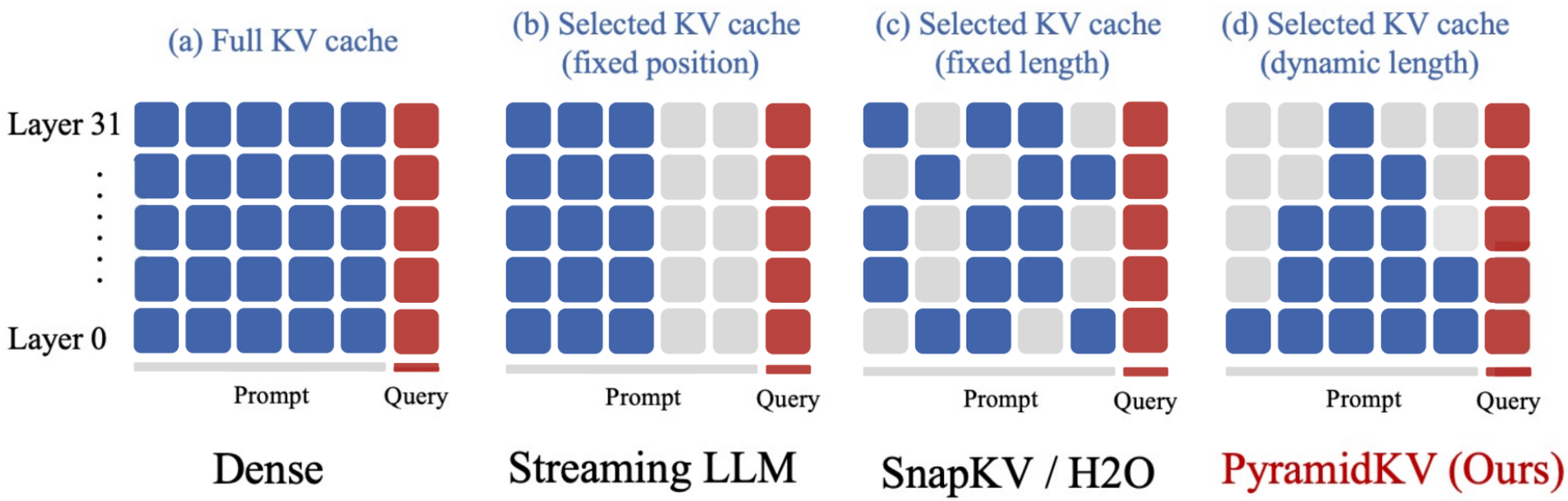


Figure 1: Illustration of PyramidKV compared with existing KV cache compression methods. (a) Full KV has all tokens stored in the KV cache in each layer; cache size increases as the input length increases. (b) StreamingLLM (Xiao et al., 2023) only keeps few initial tokens with a fixed cache size in each layer. (c) SnapKV (Li et al., 2024) and H2O (Zhang et al., 2024) keep a fixed cache size across Transformer layers, and their selection is based on the attention score. (d) PyramidKV maintains pyramid-like cache sizes, allocating more cache budget to lower layers and less to higher layers. This approach to KV cache selection better aligns with the increasing attention sparsity observed in multi-layer Transformers (§3).

Cai, et al. 2024. PyramidKV: Dynamic KV Cache Compression based on Pyramidal Information Funneling

Evaluation and Long-Context Tasks

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)

Needle-in-a-Haystack

- Multi-document QA task: retrieve the correct answer from an long text string and measure the accuracy

Input Context

```
Write a high-quality answer for the given question using only the provided search
results (some of which might be irrelevant).

Document [1](Title: Asian Americans in science and technology) Prize in physics for
discovery of the subatomic particle J/ψ. Subrahmanyan Chandrasekhar shared...
Document [2](Title: List of Nobel laureates in Physics) The first Nobel Prize in
Physics was awarded in 1901 to Wilhelm Conrad Röntgen, of Germany, who received...
Document [3](Title: Scientist) and pursued through a unique method, was essentially
in place. Ramón y Cajal won the Nobel Prize in 1906 for his remarkable...

Question: who got the first nobel prize in physics
Answer:
```

Desired Answer

```
Wilhelm Conrad Röntgen
```

Liu et al. 2023. Lost in the middle: How language models use long contexts

Needle-in-a-Haystack

- Retrieving string in a JSON object.

Input Context

```
Extract the value corresponding to the specified key in the JSON object below.

JSON data:
{"2a8d601d-1d69-4e64-9f90-8ad825a74195": "bb3ba2a5-7de8-434b-a86e-a88bb9fa7289",
 "a54e2eed-e625-4570-9f74-3624e77d6684": "d1ff29be-4e2a-4208-a182-0cea716be3d4",
 "9f4a92b9-5f69-4725-ba1e-403f08dea695": "703a7ce5-f17f-4e6d-b895-5836ba5ec71c",
 "52a9c80c-da51-4fc9-bf70-4a4901bc2ac3": "b2f8ea3d-4b1b-49e0-a141-b9823991ebeb",
 "f4eb1c53-af0a-4dc4-a3a5-c2d50851a178": "d733b0d2-6af3-44e1-8592-e5637fdb76fb"}

Key: "9f4a92b9-5f69-4725-ba1e-403f08dea695"
Corresponding value:
```

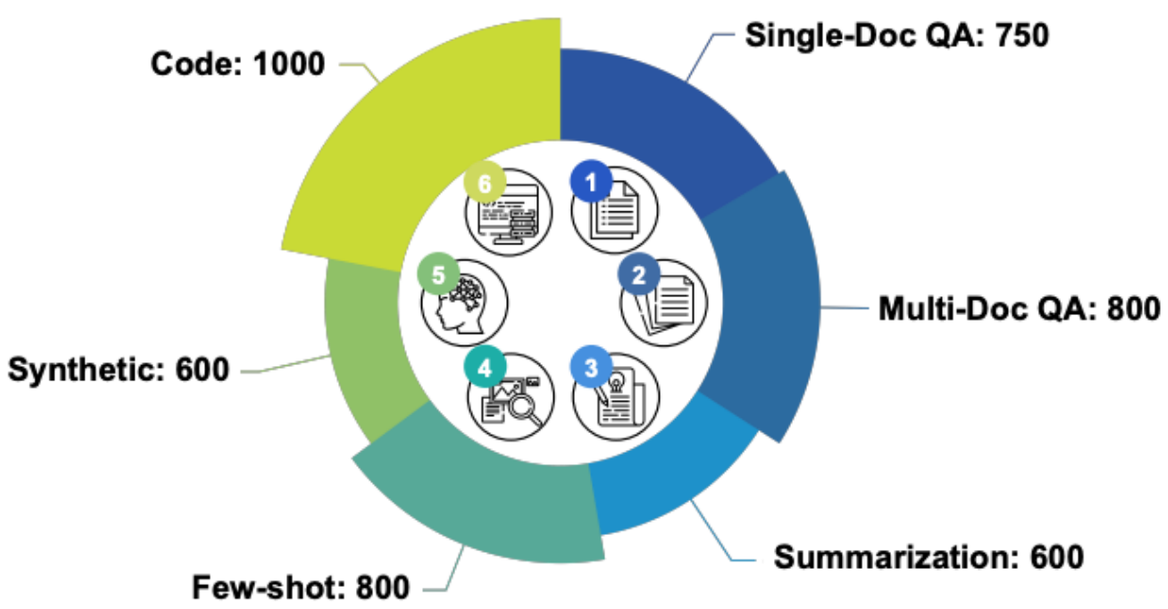
Desired Output

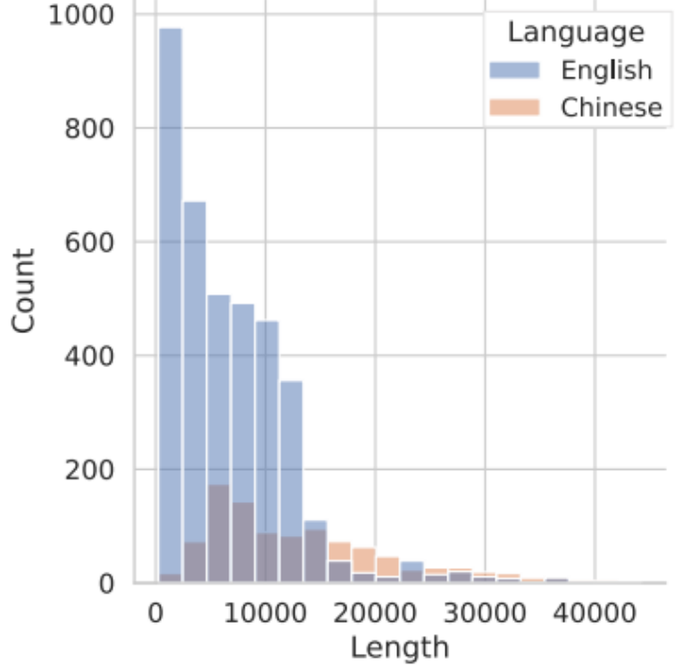
```
703a7ce5-f17f-4e6d-b895-5836ba5ec71c
```

Liu et al. 2023. Lost in the middle: How language models use long contexts

LongBench

- A bilingual, multi-task benchmark for evaluating LLMs in handling extended documents and complex information sequences.



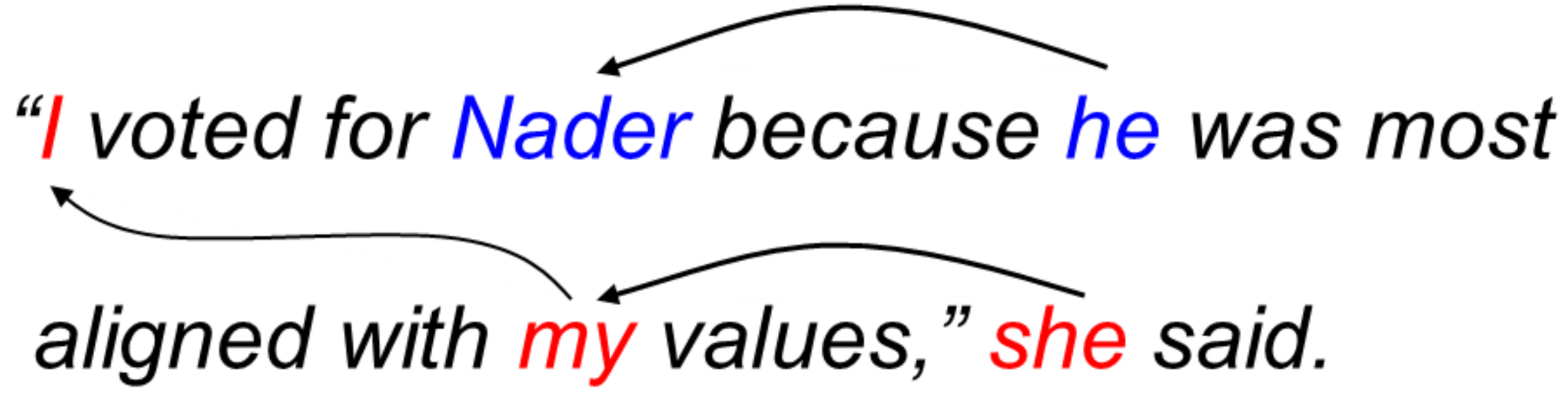


Bai, et al 2024. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding

LongBench

NarrativeQA: You are given a story, which can be either a novel or a movie script, and a question. Answer the question as concisely as you can, using a single phrase if possible. Do not provide any explanation.
Story: {context}
Now, answer the question based on the story as concisely as you can, using a single phrase if possible. Do not provide any explanation.
Question: {input}
Answer:

QMSum: You are given a meeting transcript and a query containing a question or instruction. Answer the query in one or more sentences.
Transcript:
{context}
Now, answer the query based on the above meeting transcript in one or more sentences.
Query: {input}
Answer:



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).

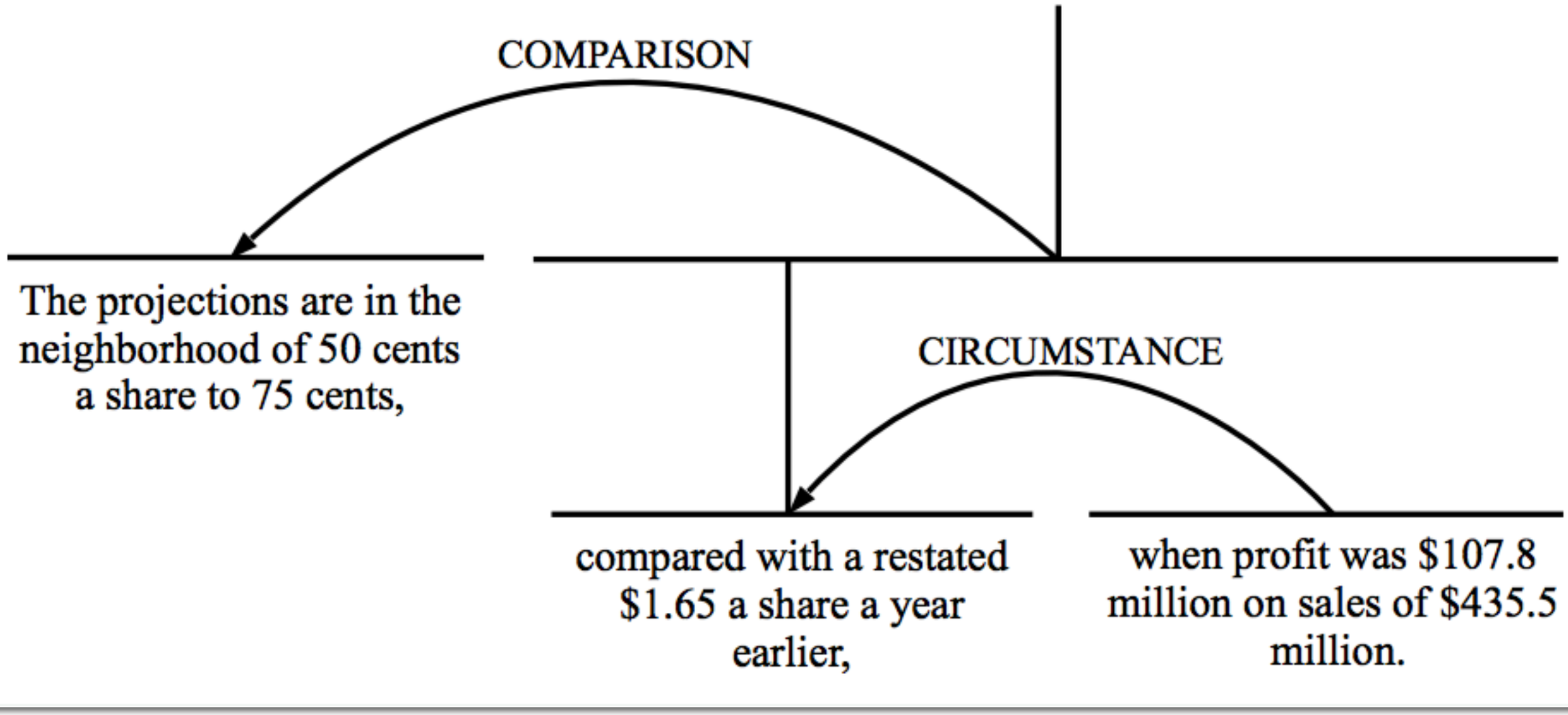
- Step 2: Cluster noun phrases (**mentions**) referring to the same underlying world **entity**.

Mention(Noun Phrase) Detection

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

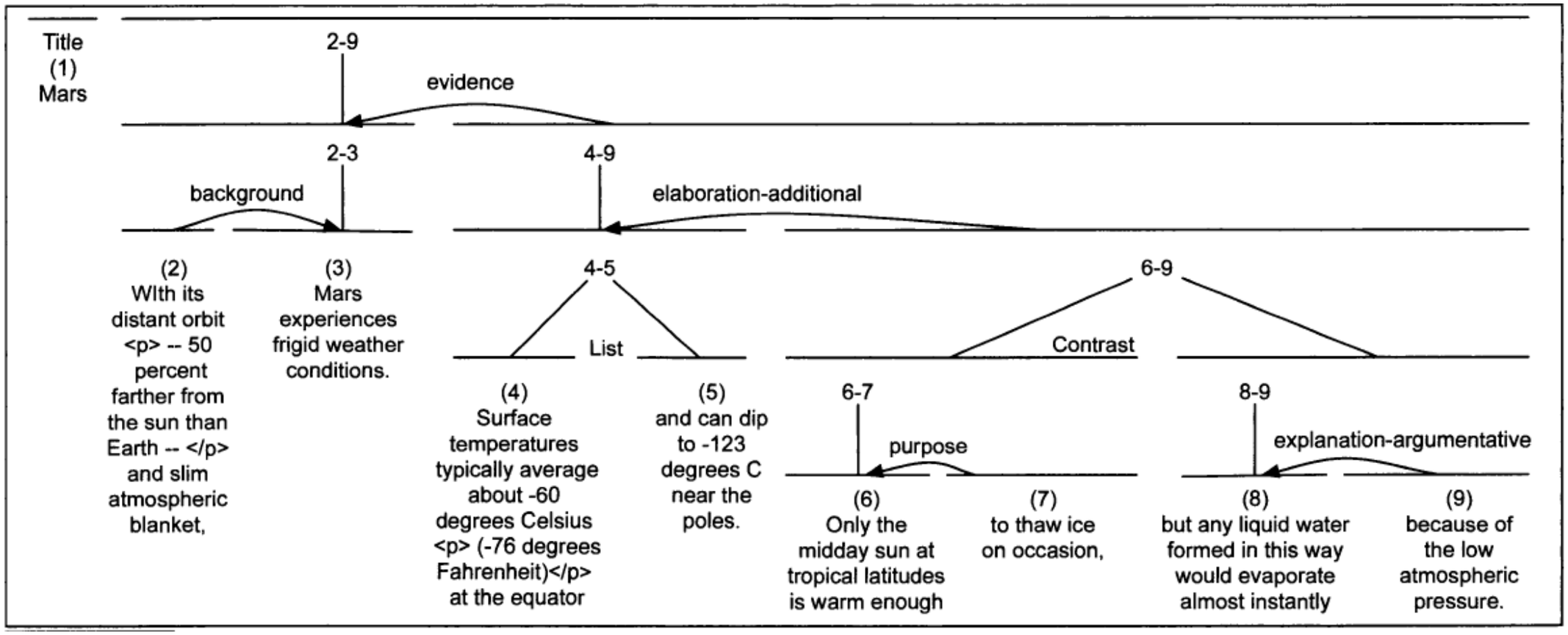
A renowned speech therapist was summoned to help the King overcome his *speech impediment*...

- 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.



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)

Questions?