Pre-Training & In-context Learning

Large Language Models: Introduction and Recent Advances

ELL881 · AIL821

Gaurav Pandey
Research Scientist, IBM Research

How to make ChatGPT ?

- Pre-Training
  - This is the point where most of the reasoning power is infused in the model.
  - Data – Billions of tokens of unstructured text from the internet
- Instruction Tuning
  - Trains models to follow natural language instructions
  - Data – Several thousand (Task, Instruction, Output) triplets
- Reinforcement Learning from Human Feedback
  - Show the output(s) generated by models to humans/reward model
  - Collect feedback in the form of preferences.
  - Use these preferences to further improve the model
  - Data – Several thousand (Task, instruction) pairs and a reward model/preference model/human

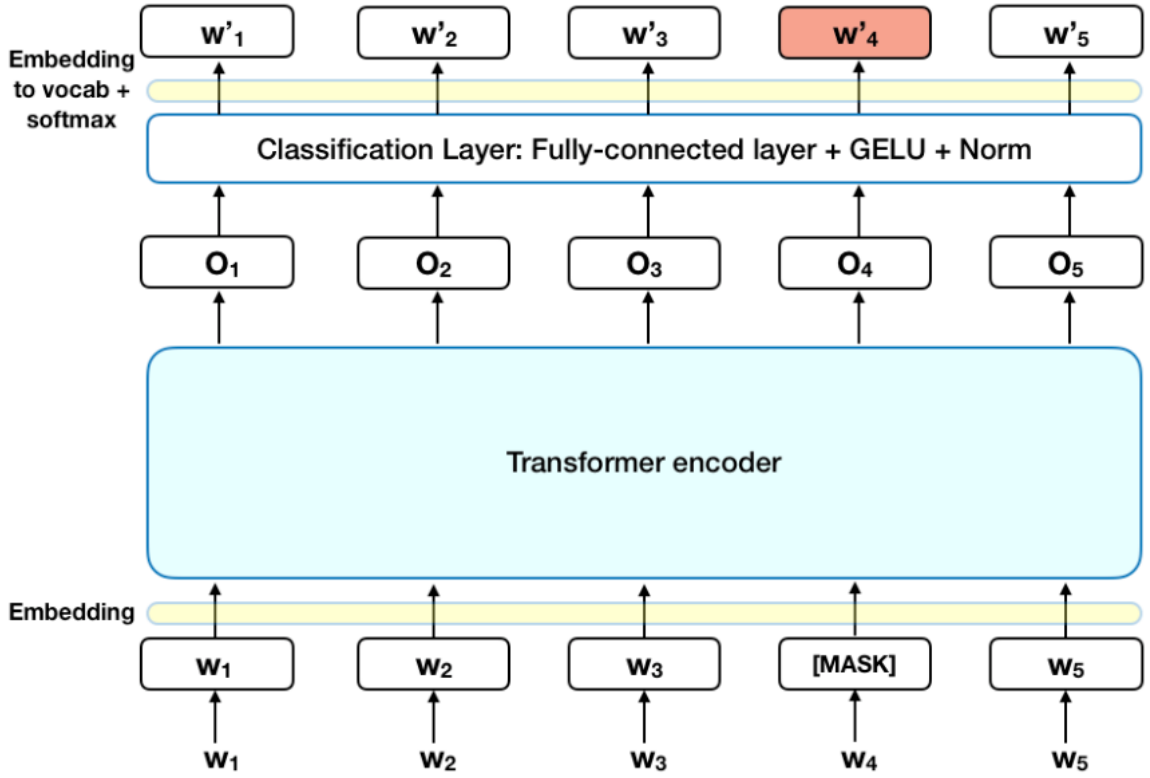
Why Do We Need Pre-Training?

- Humans Excel at Generalization from Limited Examples
  - Eg: C-A-T → T-A-C D-O-G → ??
  - Lifetime of accumulated knowledge to leverage from.
- Traditional ML requires vast amount of data for each task
  - Need to learn each pattern from scratch
  - No prior knowledge to guide them
- **Question:** Can we somehow use the vast amount of data on the web to
  - Achieve better performance with less data?

A generative model for sentences

How to pretrain?

- Masked language models



Not a generative model !!
BERT can be used for learning sentence embeddings but not for generating sentences

Image courtesy: Rani Horev

Generative modelling of sentences

- **What do we need** – A model from which we can sample sentences
- Given a sentence $s = (t_1, \ldots, t_m)$, the probability of the sentence can be written as

$$p(s) = p_1(t_1) \prod_{j=1}^{m} p_{j+1}(t_{j+1} | t_1, \ldots, t_j)$$

→ Autoregressive

$p(\text{token1})$ $\quad$ $p(\text{token2} \mid \text{token1})$ $\times$

- p_j are distributions over the vocabulary.
- If these p_js are provided, such a model is called autoregressive model

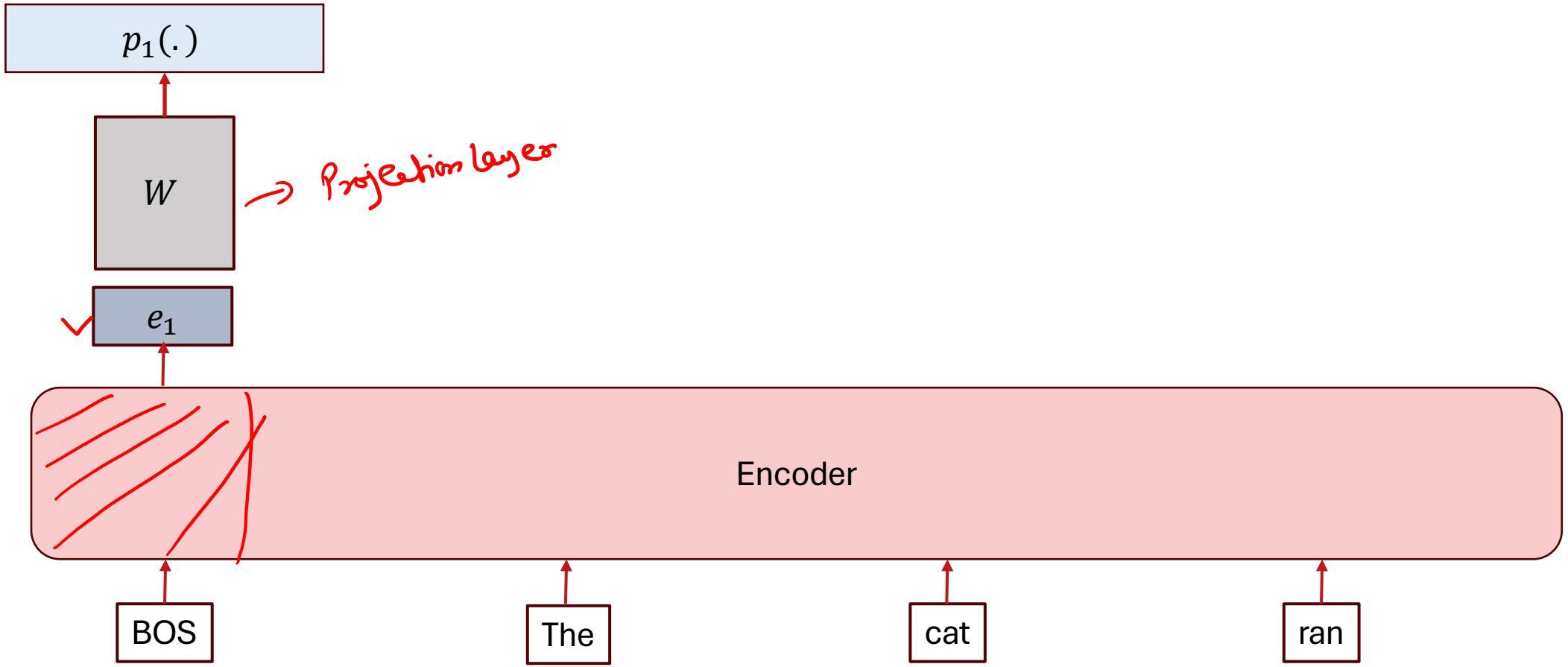
Sampling a sentence from an autoregressive model

- Sample the first token t_1 from p_1
- For each subsequent step
  - Sample token t_{j+1} conditioned on the previous tokens $t_1, \ldots, t_j$

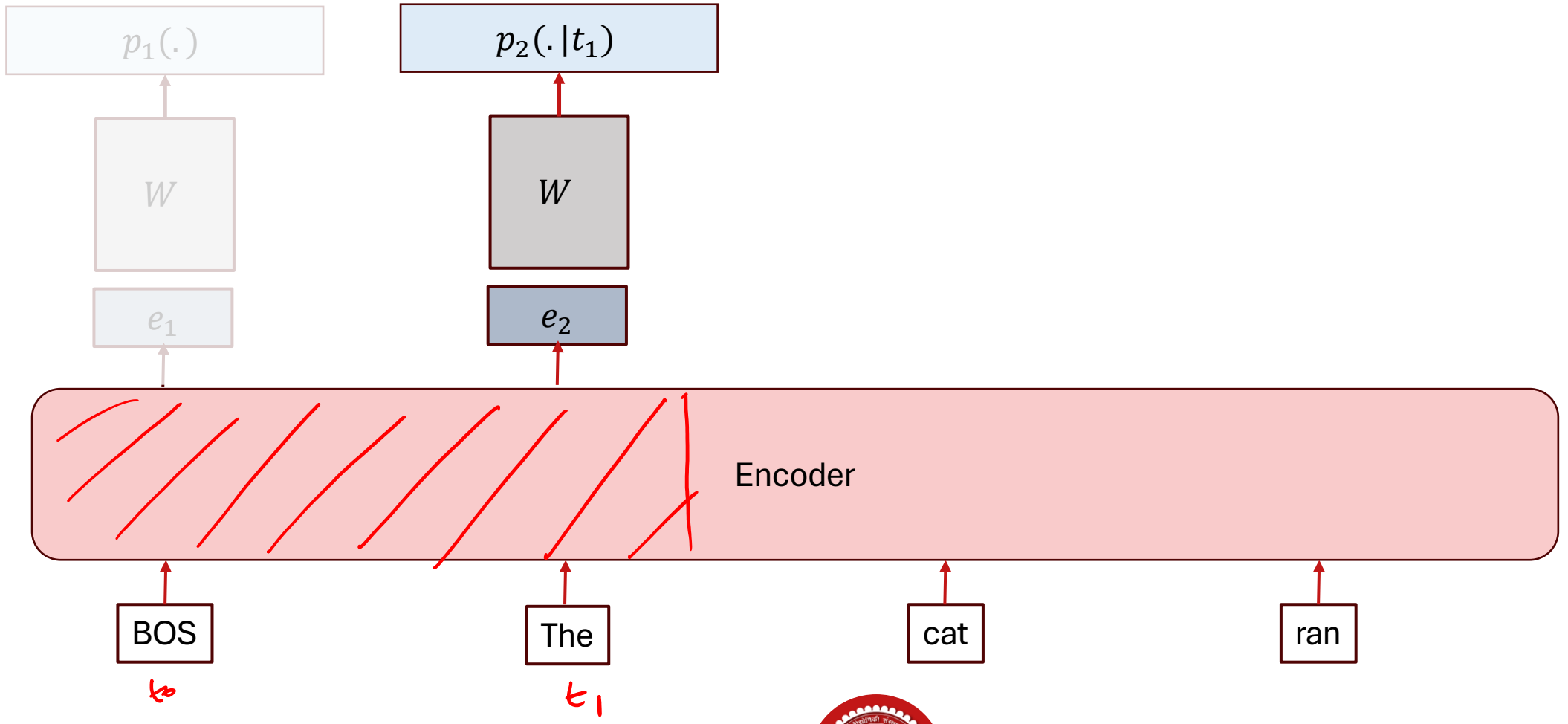
$$t_{j+1} \sim p_{j+1}(.|t_1, \ldots, t_j)$$

Since p_j are distributions over the vocabulary, they are easy to sample from !!

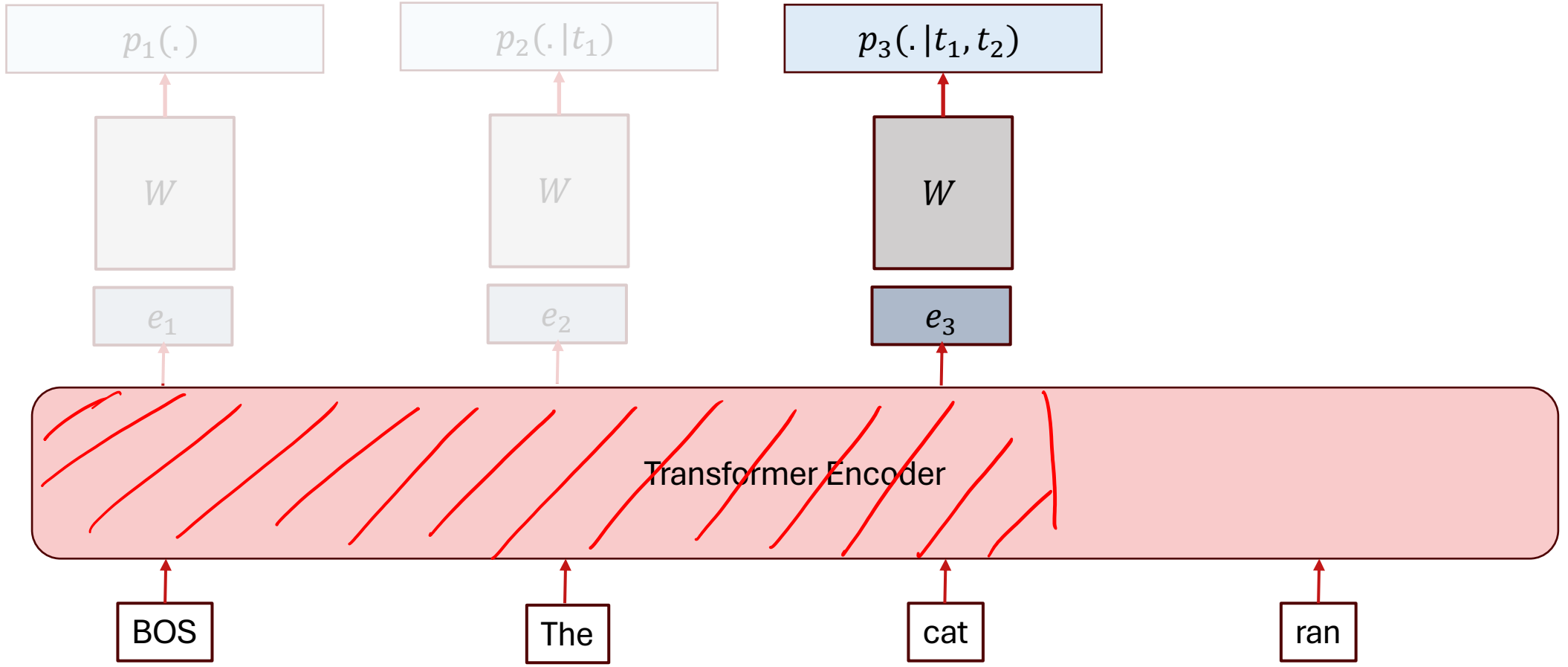
What do p_is look like?



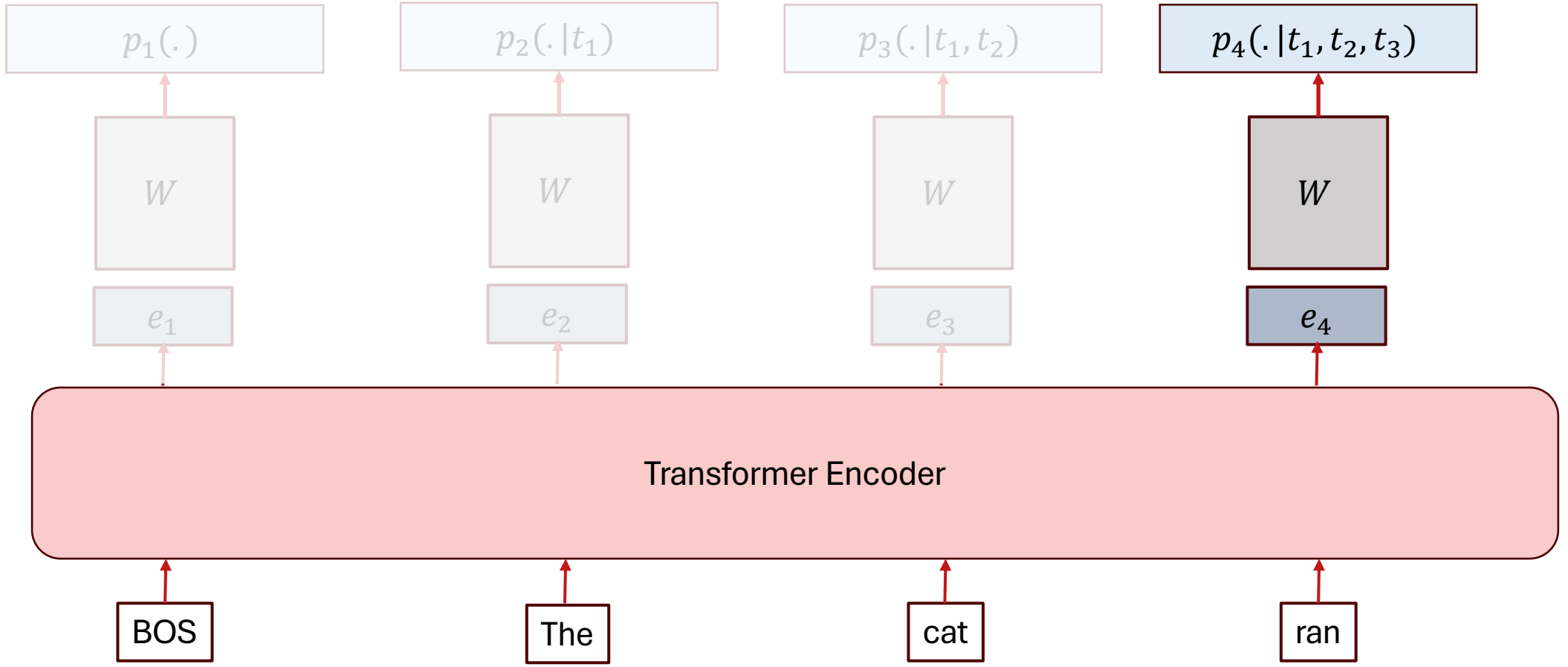
What do p_is look like?



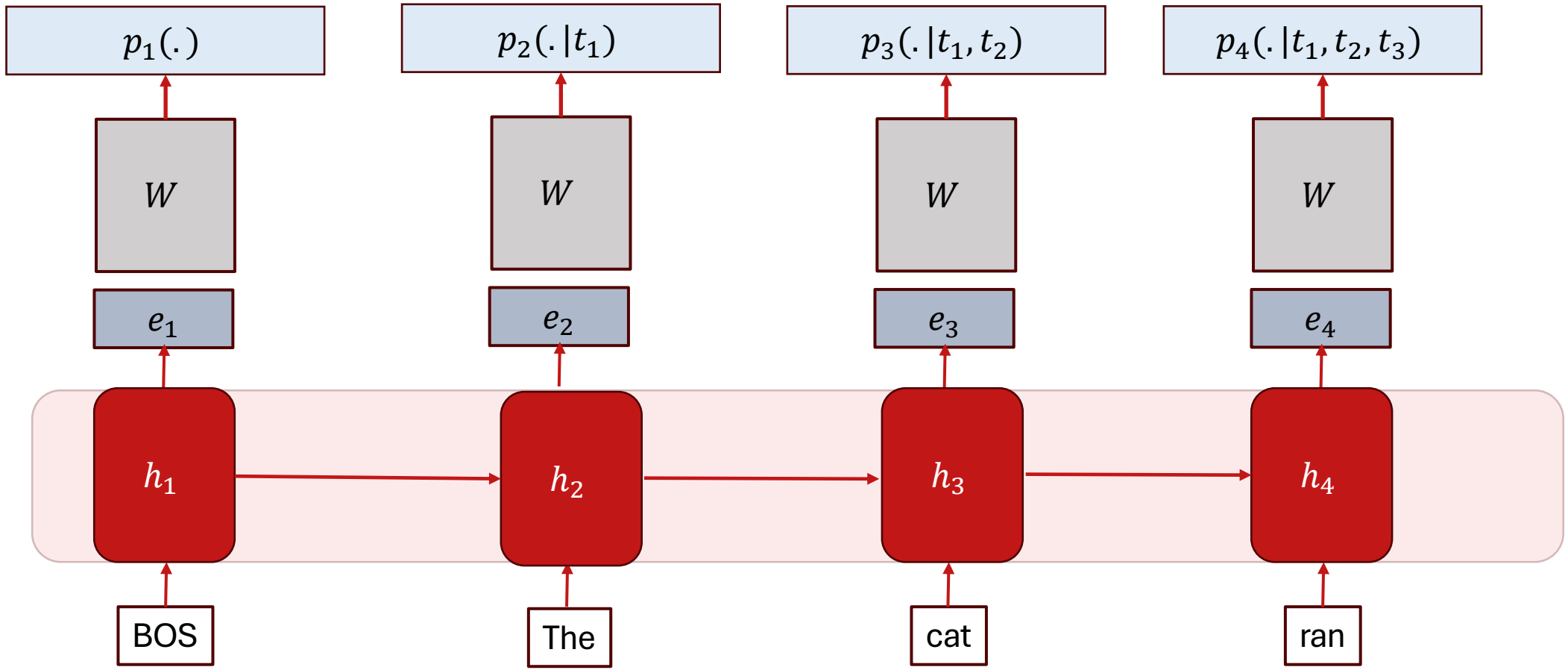
What do p_is look like?



What do p_is look like?



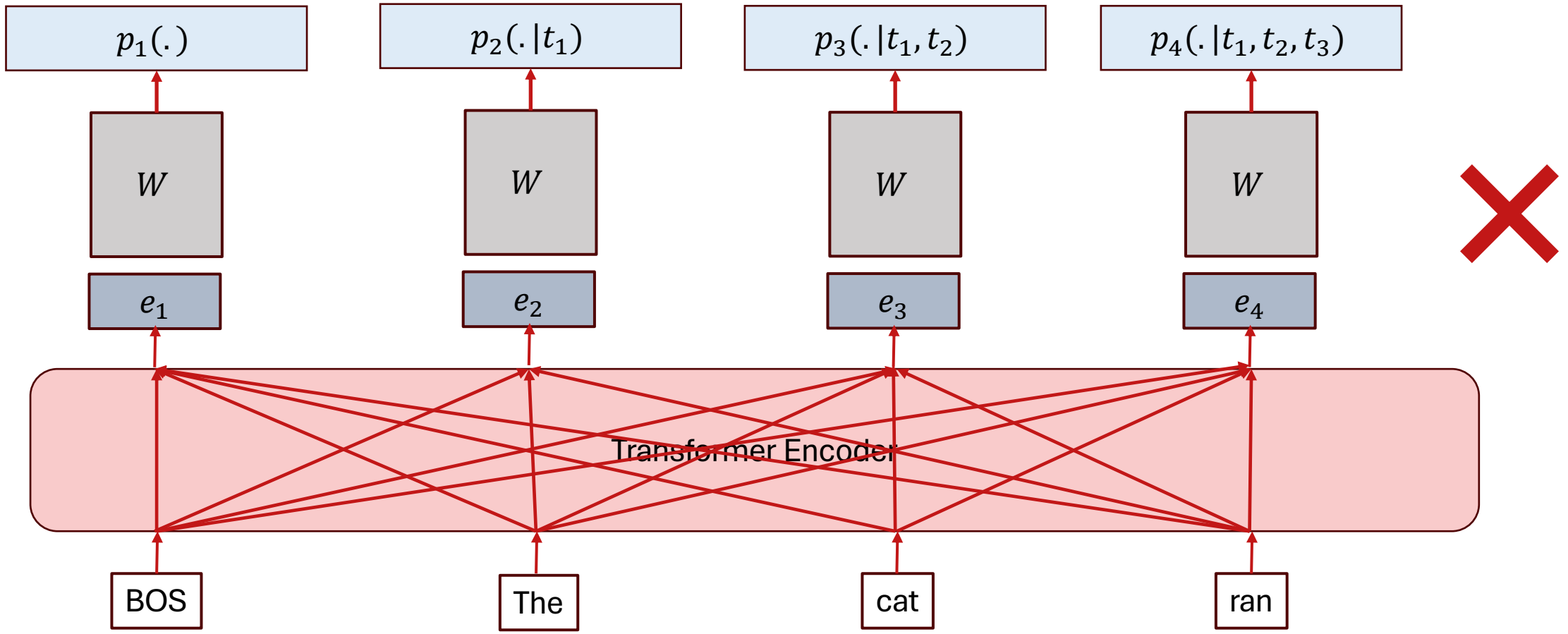
What do p_is look like? – RNN encoder



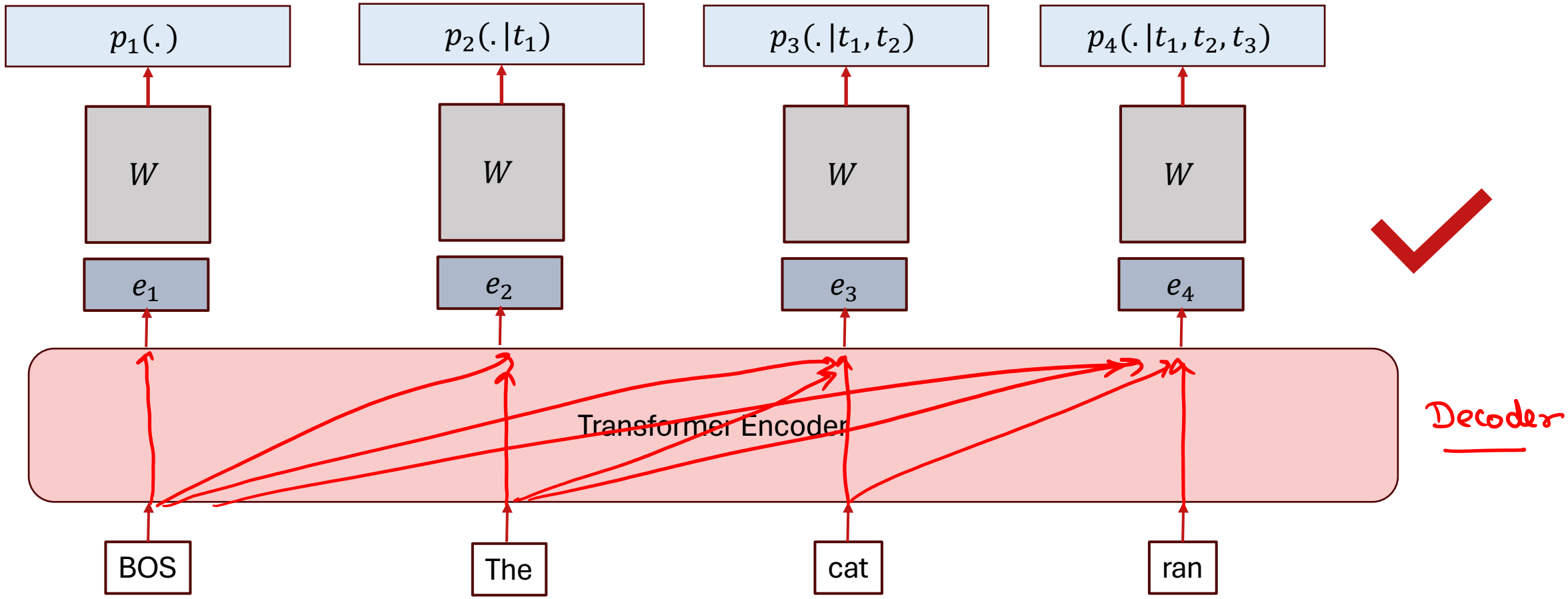
Problems with RNN Encoder

- The hidden state at $(j+1)^{th}$ step depends on the hidden step at j^{th} step.
  - $h_{j+1} =$ $f(h_j, t_j)$
  - Time $\propto$ No. of tokens.
- How to get rid of sequential computation ?

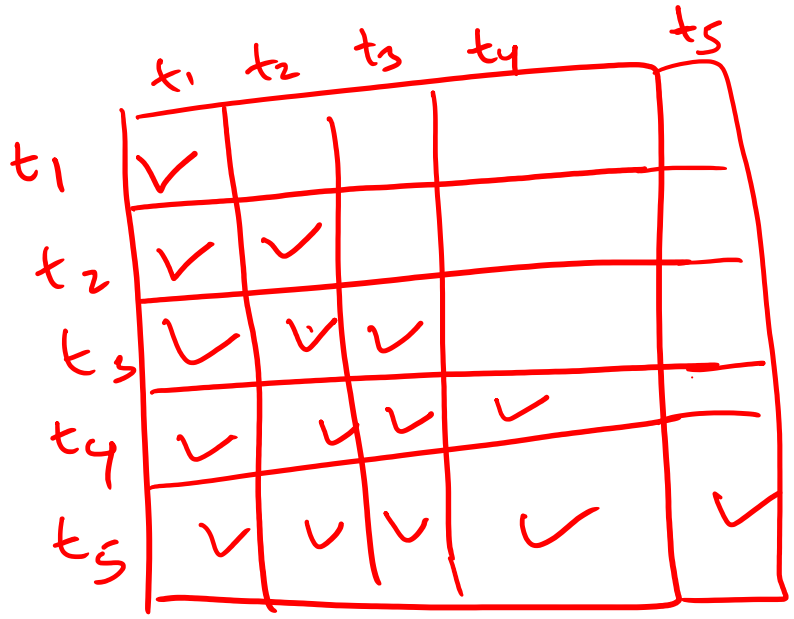
What do p_is look like? – Full attention mask



What do p_is look like? – Causal attention mask



Causal attention mask



Training Loss – Recap: Maximum Likelihood Estimation

- Given
  - A parametric family of probability distributions p_θ
  - n independent & identically distributed observations $s_1, \ldots, s_n$ ~ p_θ
- Task
  - Find the parameter θ^* that most likely resulted in the observed data, that is

$$\begin{aligned}
&\arg\max_\theta p_\theta(s_1, \ldots, s_n) \quad \Rightarrow \text{Joint prob distn} \\
&= \arg\max_\theta \log p_\theta(s_1, \ldots, s_n) \\
&= \arg\max_\theta \sum_{i=1}^{n} \log p_\theta(s_i)
\end{aligned}$$

MLE estimation from sentence data

- Given a sentence $s = (t_1, \ldots, t_m)$, the probability of the sentence can be written as

$$p_\theta(s) = p_1(t_1) \prod_{j=1}^{m} p_{j+1}(t_{j+1}|t_1, \ldots, t_j)$$

$$\log p_\theta(s) = \log p_1(t_1) + \sum_{j=1}^{m} \log p_{j+1}(t_{j+1}|t_1, \ldots, t_j)$$

- Maximum Likelihood estimation

$$\arg\max_\theta \frac{1}{|\mathcal{D}|} \sum_{s \in \mathcal{D}} \log p_\theta(s)$$

Pre-Training

Pre-Training Considerations

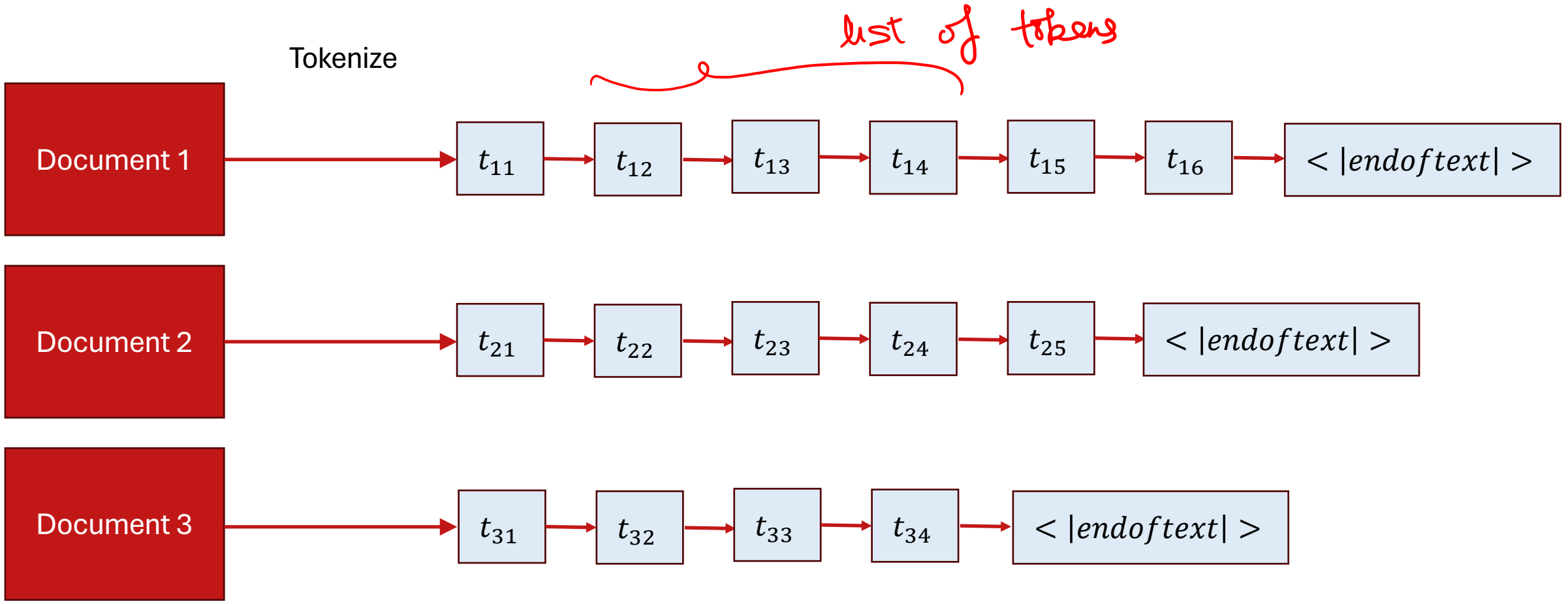
- Pre-Training Data
  - Large scale web corpus is curated & filtered
- Tokenization – Tokenizer & Vocabulary size ~ 50 000
- Model architecture
  - Number of layers
  - Token representation dimension/hidden dimension
  - Number of attention heads
  - Maximum sequence length → inference / pretraining
- Optimizer
  - Adam/Adagrad/Adafactor
  - Hyperparameters – learning rate, beta-values

Data curation & filtering

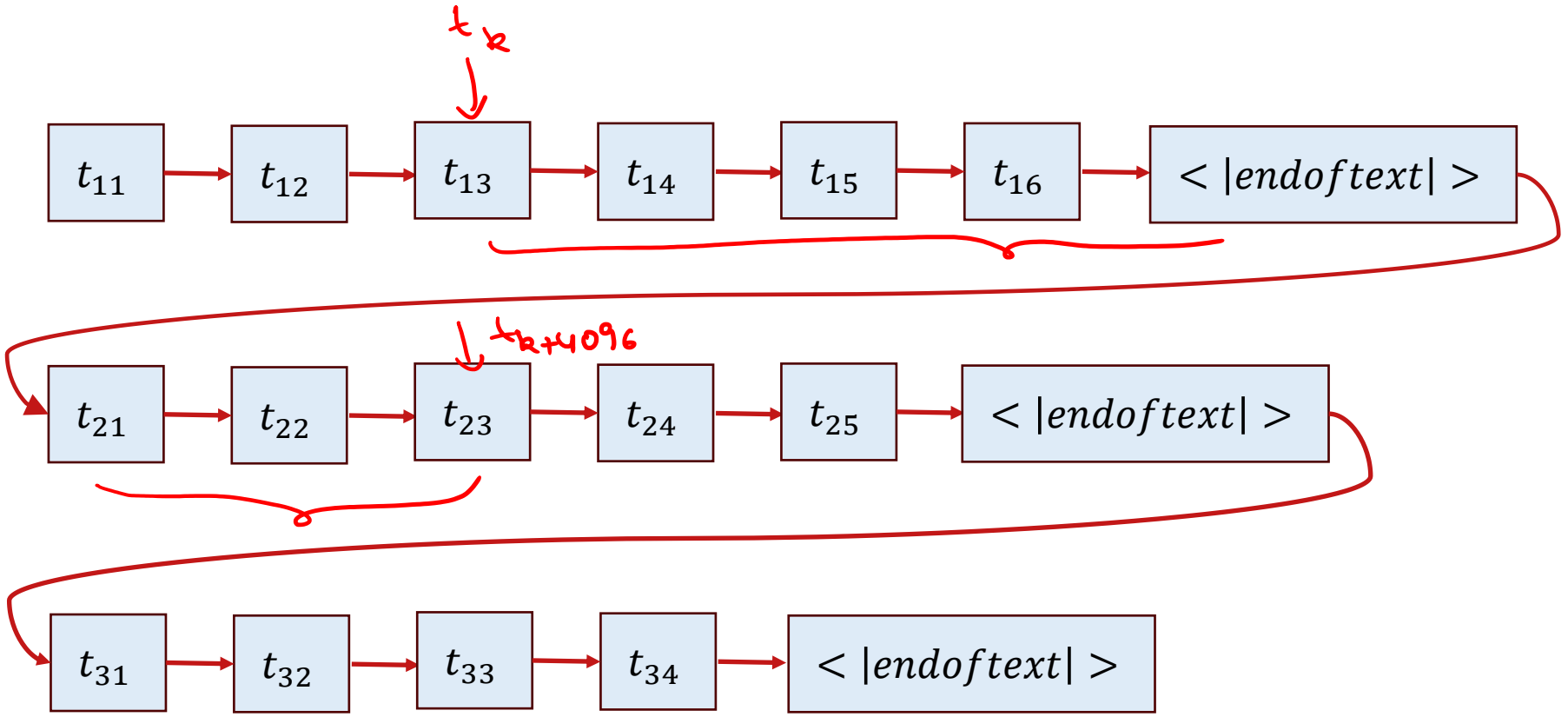
- Scrape the web for HTML/text documents
- Parse the HTML web page using HTML parsers
  - Removal of html tags
- Deduplication of web pages
- Remove non-informative documents – logs & error messages
- Remove outlier documents – documents with weird token distribution
- **Optional**: Train a classifier to determine document quality

Credit: Llama 3.1 paper by Meta

Tokenization & Data stream



Tokenization & Data stream



No padding is used during pretraining

What can pre-trained models do?

- They can mimic the behavior that they saw on the text on which they were trained on.

”I’m not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I’m not a fool].**

In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: ”**Mentez mentez, il en restera toujours quelque chose**,” which translates as, ”**Lie lie and something will always remain.**”

“I hate the word ‘**perfume**,”’ Burr says. ‘It’s somewhat better in French: ‘**parfum**.’

If listened carefully at 29:55, a conversation can be heard between two guys in French: “**-Comment on fait pour aller de l’autre coté? -Quel autre coté?**”, which means “**- How do you get to the other side? - What side?**”.

If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?

“**Brevet Sans Garantie Du Gouvernement**”, translated to English: “**Patented without government warranty**”.

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

Here is a poem that describes a sun as it gets dipped in chocolate.

It is a beautiful day,
The sun is shining bright,
And the chocolate is melting,
As it gets dipped in the night.

Input: Prompt and the first line

Model: Llama-3-8B

What can they not do?

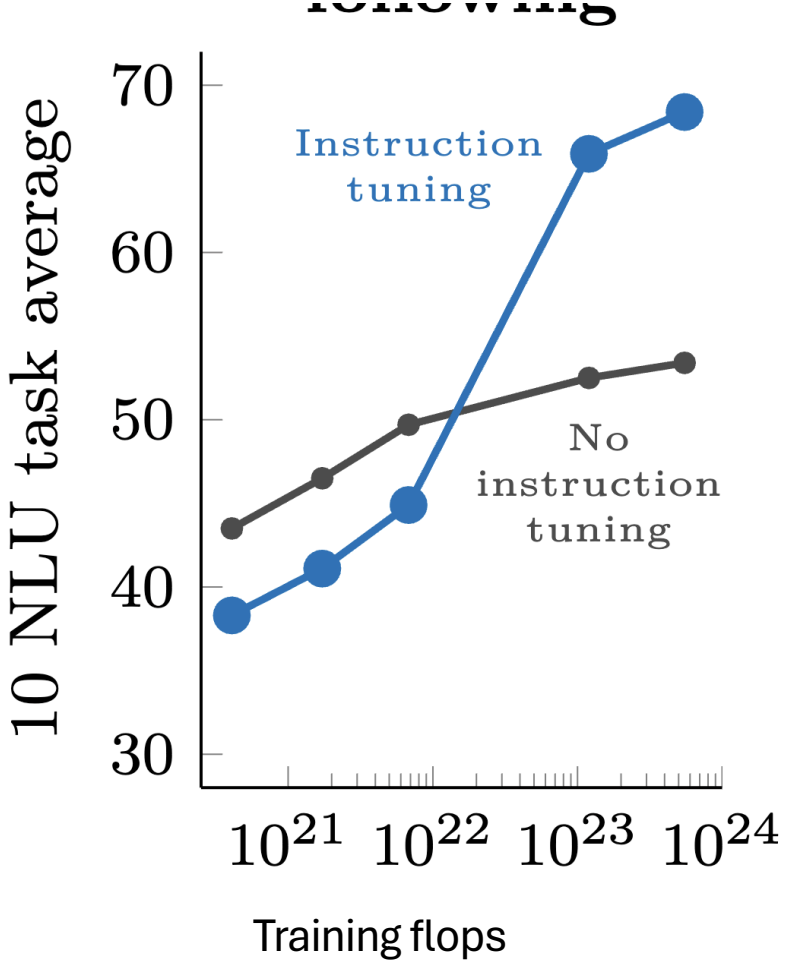
Write a poem that describes a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should be about a sun getting dipped in chocolate. It should be a haiku, and it should

They can’t follow instructions !!

Reason: Most of their training data is not in instruction-output format

How to make them work ?

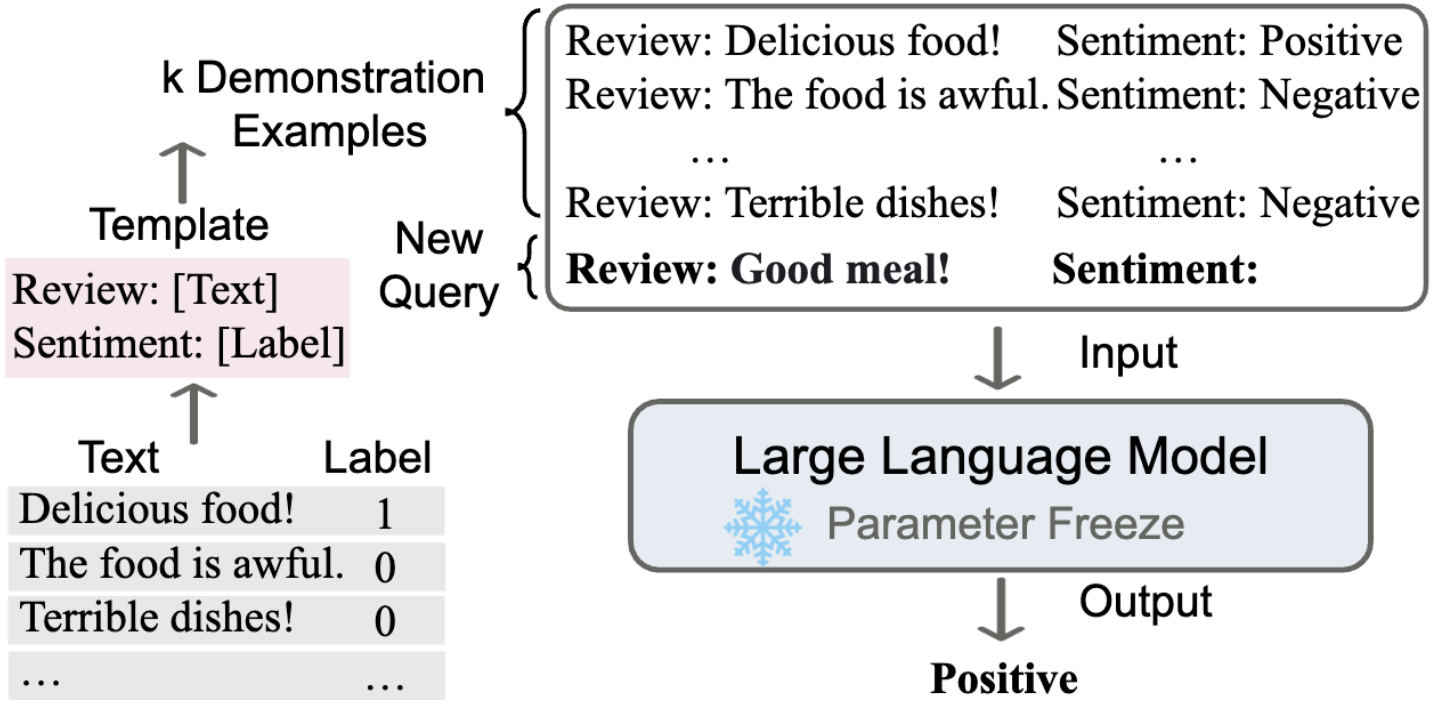
- In-context learning (this lecture)
  - Give few examples of the task that you want the model to solve
- Instruction tuning (next lecture)
  - Finetune the pretrained model to follow instructions



Emergence of In-context learning

What is in-context learning?

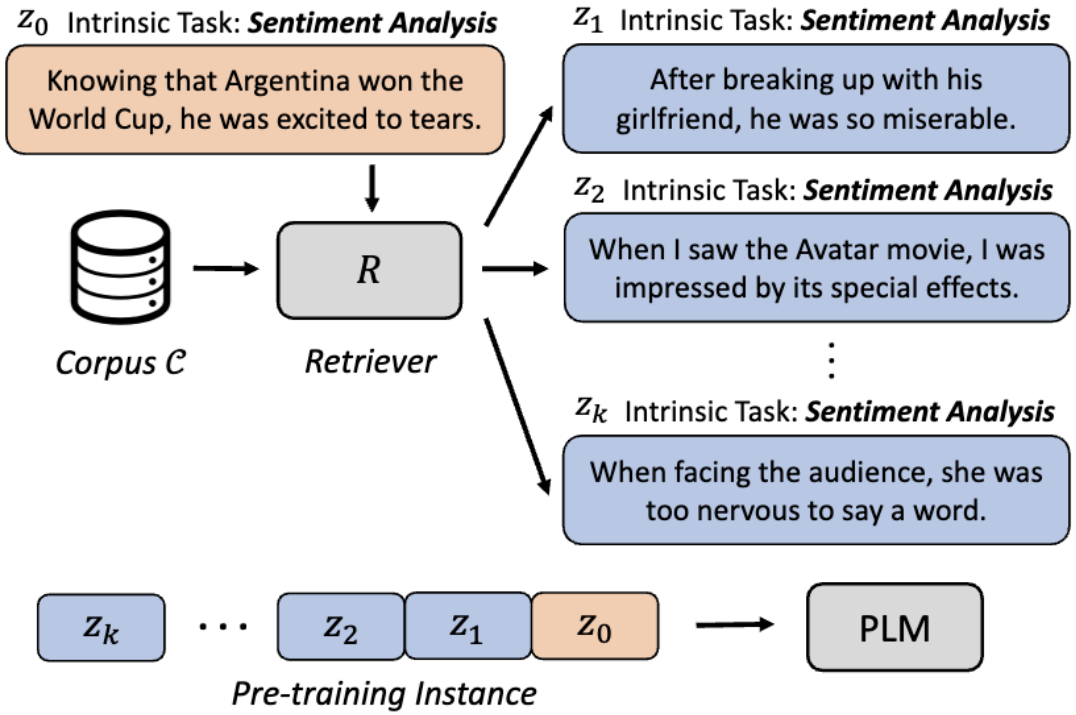
- Learning from a few examples within the context



Credit: A Survey on In-context Learning

Boosting In-context learning - during pre-training

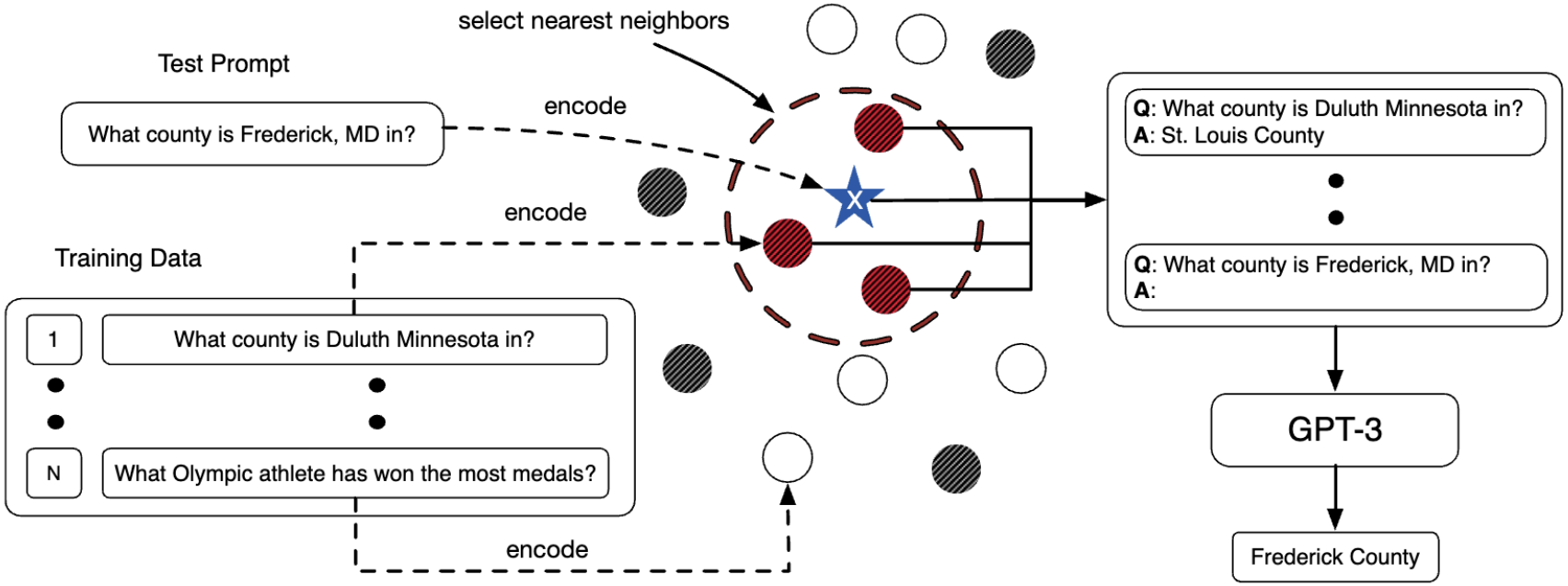
- Reorganize pretraining data to group similar examples together



Credit: Pre-Training to Learn in Context

Boosting In-context learning – during inference

- Choose in-context examples similar to the query example



Credit: What Makes Good In-Context Examples for GPT-3

What is a good similarity metric?

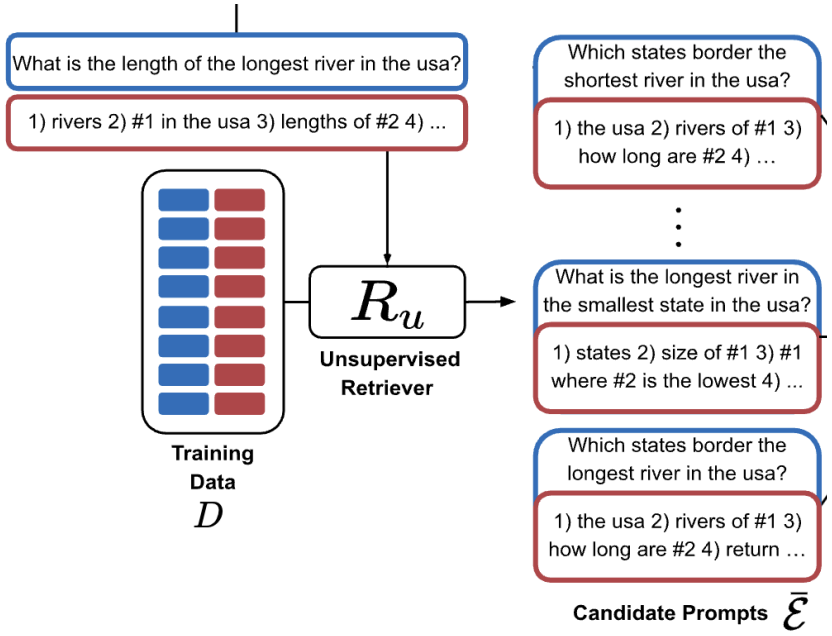
- Cosine distance
  - Embed the inputs using a sentence embedder such as Roberta.
  - Similar examples are those whose inputs maximize the cosine similarity.
  - Given query q, the score of an example (x, y) is given by

$$Score(x) = < \frac{e(x)}{||e(x)||}, \frac{e(q)}{||e(q)||} >$$

  - The top-k examples are selected.

- Question: How can you make this set diverse?

Contrastive learning of the similarity metric

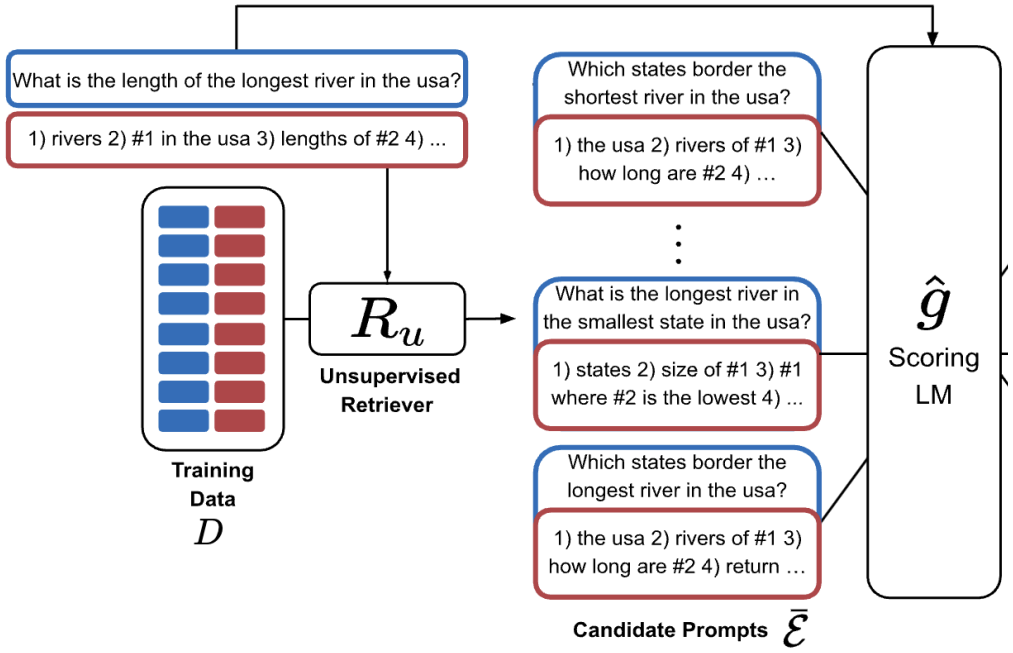


$$s(x_1,y_1) = \log p(y \mid x, (x_1,y_1))$$

$$s(x_2,y_2) = \log p(y \mid x, (x_2,y_2))$$

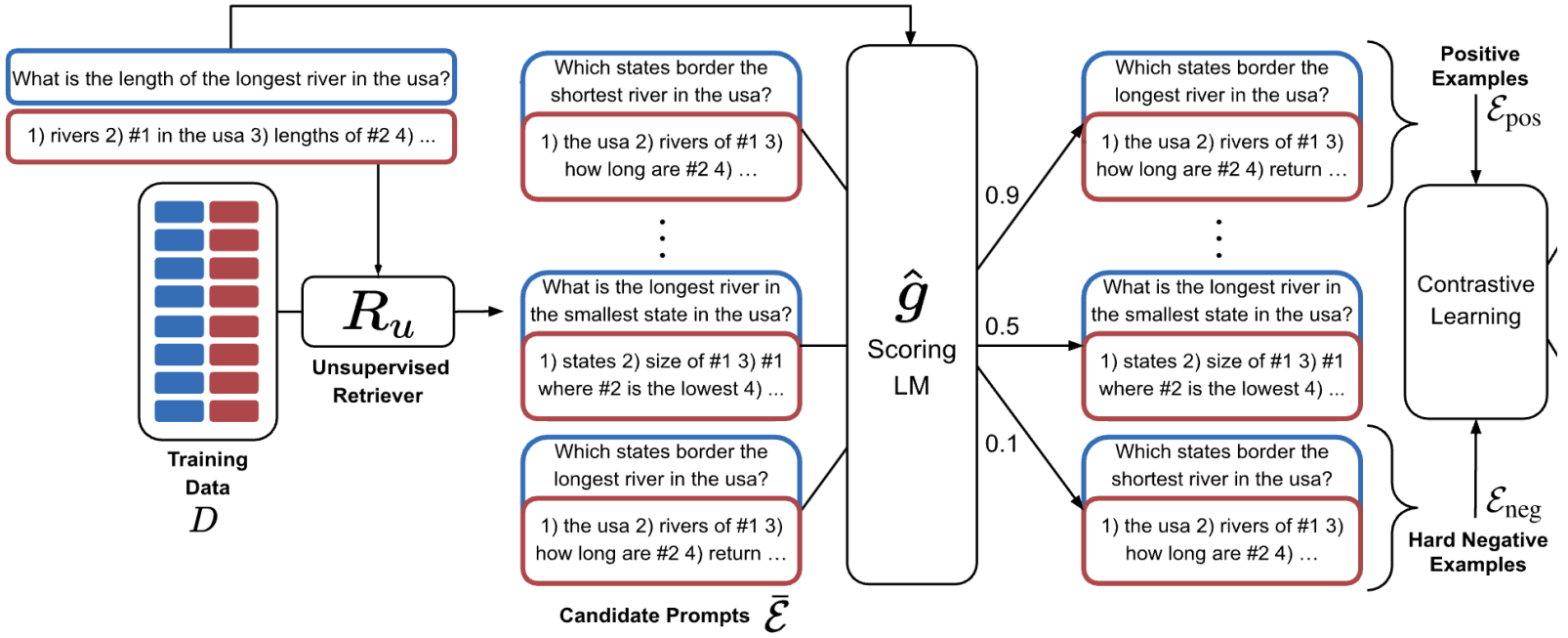
Credit: Learning To Retrieve Prompts for In-Context Learning

Contrastive learning of the similarity metric



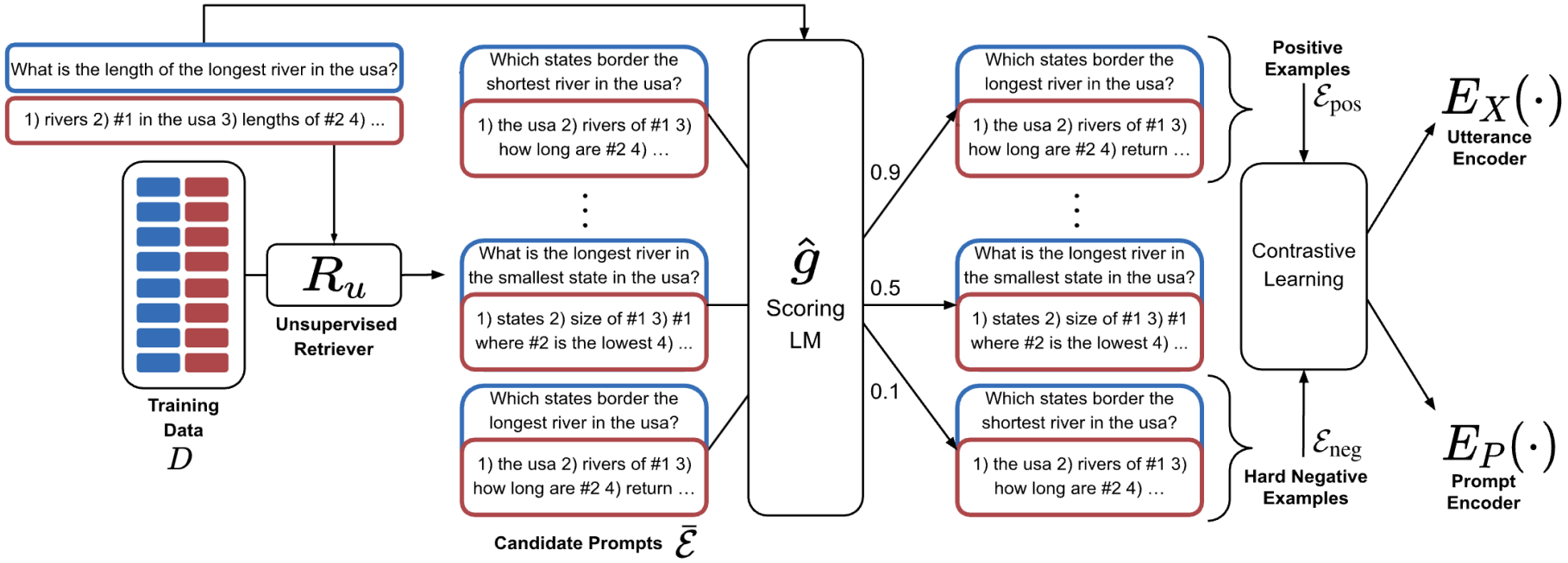
Contrastive learning of the similarity metric

$e_1 = (x_1, y_1)$ had the highest
$e_n = (x_n, y_n)$ → lowest
$e(x_1) \equiv e(x)$
$e(x_n)$ far from x
triplet loss



Credit: Learning To Retrieve Prompts for In-Context Learning

Contrastive learning of the similarity metric



Credit: Learning To Retrieve Prompts for In-Context Learning

Why does in-context learning work?

- Motivation:

Prompt: (apples are red) (bananas are yellow) (grapes are ??)

GPT-3: (apples are red) (bananas are yellow) (grapes are purple)

Prompt: (lemons are sour) (cranberries are bitter) (grapes are ??)

GPT-3: : (lemons are sour) (cranberries are bitter) (grapes are sweet)

Claim 1: Transformers implicitly perform Bayesian inference.
Claim 2: Transformers learn in-context by gradient descent !!

Content credit: **TILOS Seminar: Transformers learn in-context by (functional) gradient descent**

Recap – Gradient Descent

$x_1: (0.1, 0.3, -0.7)$, $y_1: -1.2$
$x_2: (0.2, 0.4, -0.6)$, $y_2: -2.4$
.....
$x_n: (0.4, 0.7 \ -0.3)$, $y_n: -1.6$

Assumption: Linear model

$$y =< \theta, x >$$

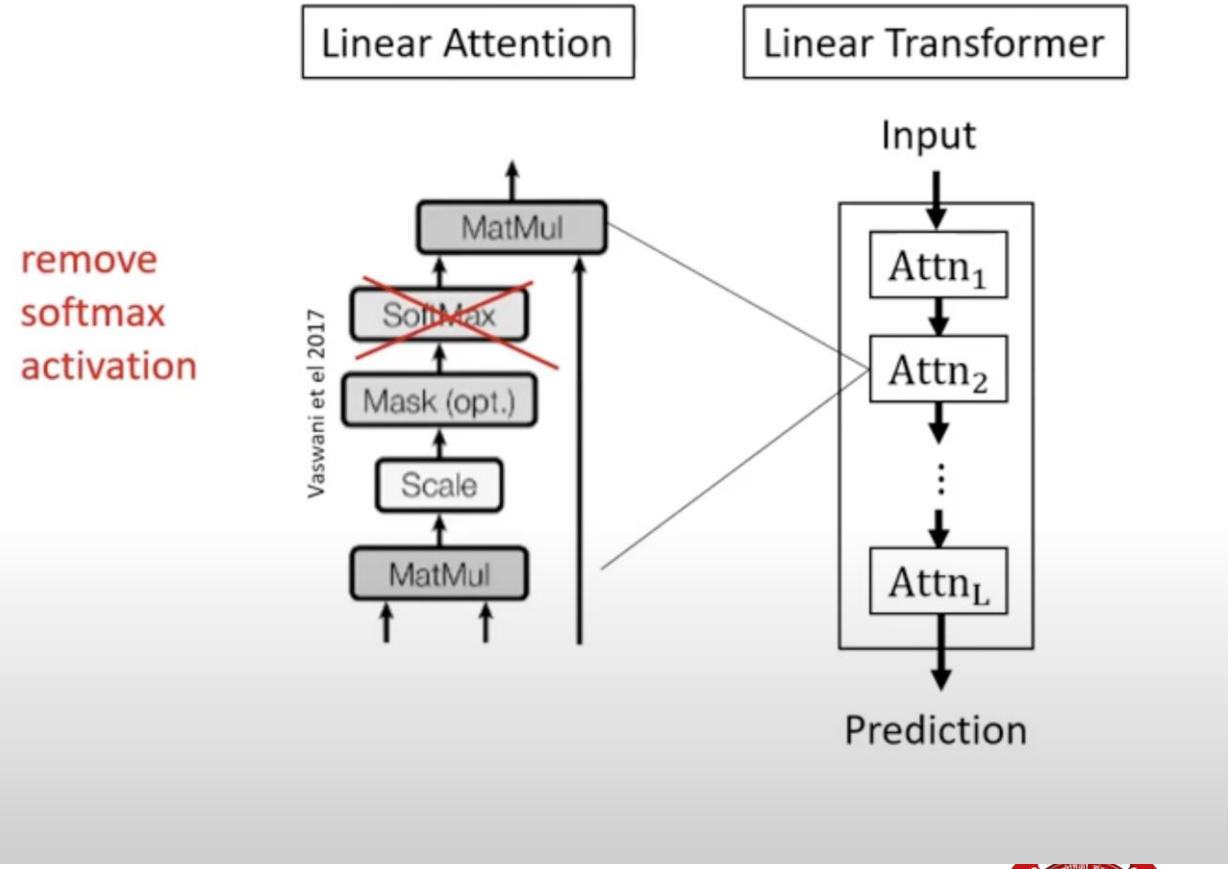
Loss $R(\theta) = \frac{1}{n} \sum_{i=1}^{n} | y_i - \theta^T x_i |^2$.

Gradient Descent

$$\theta_{t+1} = \theta_t - \delta_t \nabla_\theta R(\theta_t)$$

Content credit: **TILOS Seminar: Transformers learn in-context by (functional) gradient descent**

Linear transformers

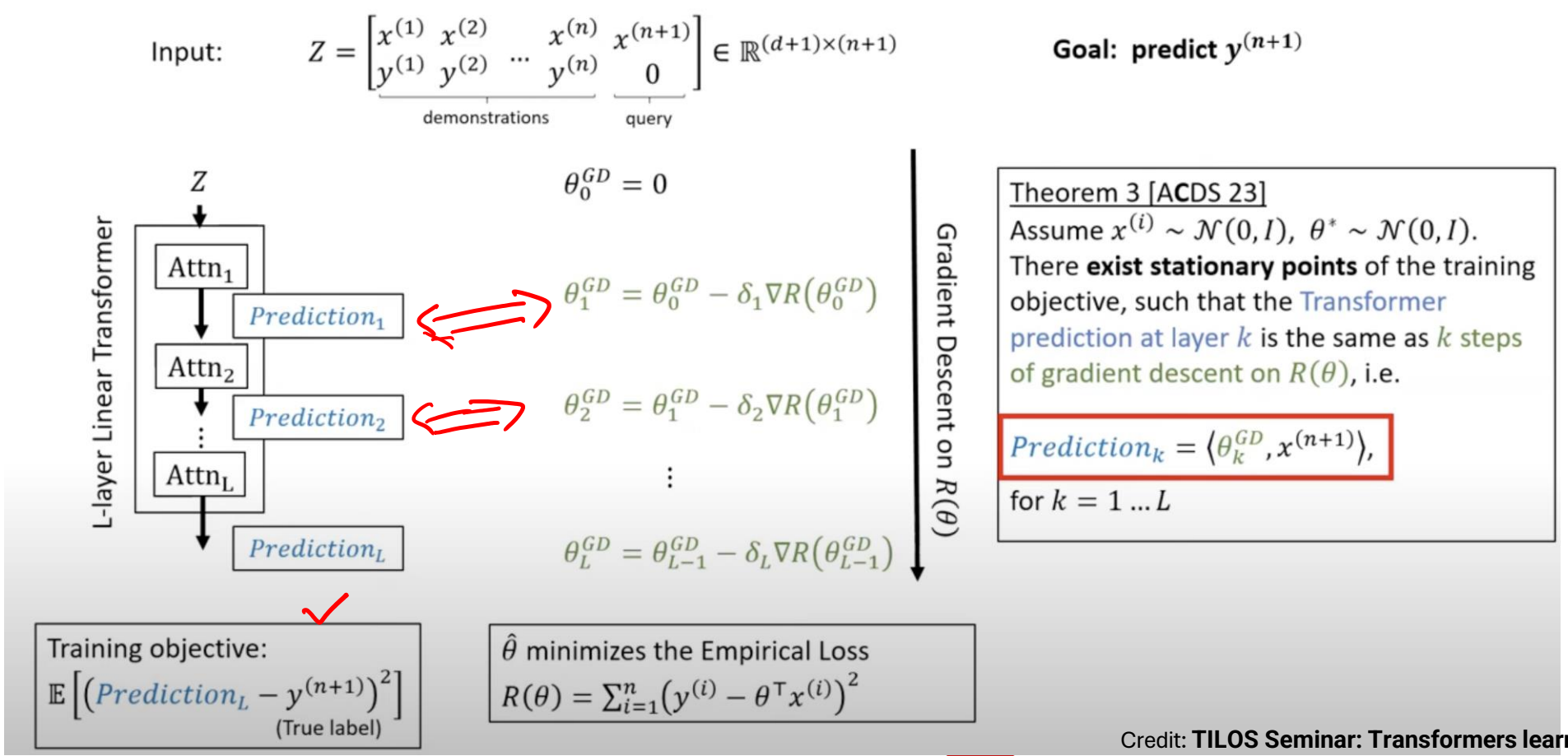


$$\cancel{\text{Softmax}}\left(\frac{QK^T}{\sqrt{d}}\right)V$$

$$\left(\frac{QK^T}{\sqrt{d}}V\right)$$

Credit: **TILOS Seminar: Transformers learn in-context by (functional) gradient descent**

Transformers learn in-context by gradient descent



Credit: **TILOS Seminar: Transformers learn in-context by (functional) gradient descent**

Main Takeaways

- A generative model for sentences can be created by using a causal attention mask in transformers.
- In-context learning (ICL) is an emergent property of pre-trained models.
- ICL performance can be improved during
  - Pretraining
    - Diversity of pretraining corpora significantly improves in-context learning performance.
    - Clustered data is good for ICL – similar examples should occur together.
  - Inference
    - Examples with embeddings closer to the query samples yield better performance.
    - Diversity among examples yields better performance.
- In-context learning implicitly performs some form of gradient descent.