

Pre-training of Causal LMs & 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?

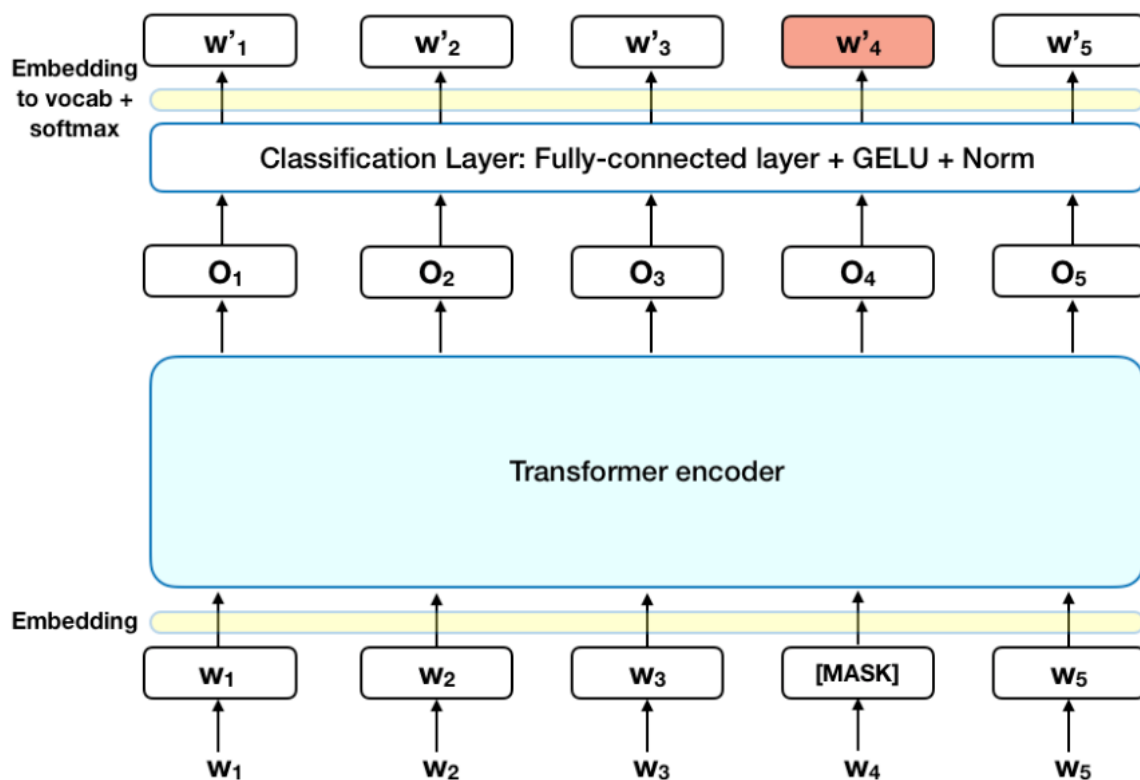


A generative model for sentences



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 $s = (t_1, \dots, t_m)$, the probability of the sentence can be written as

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

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



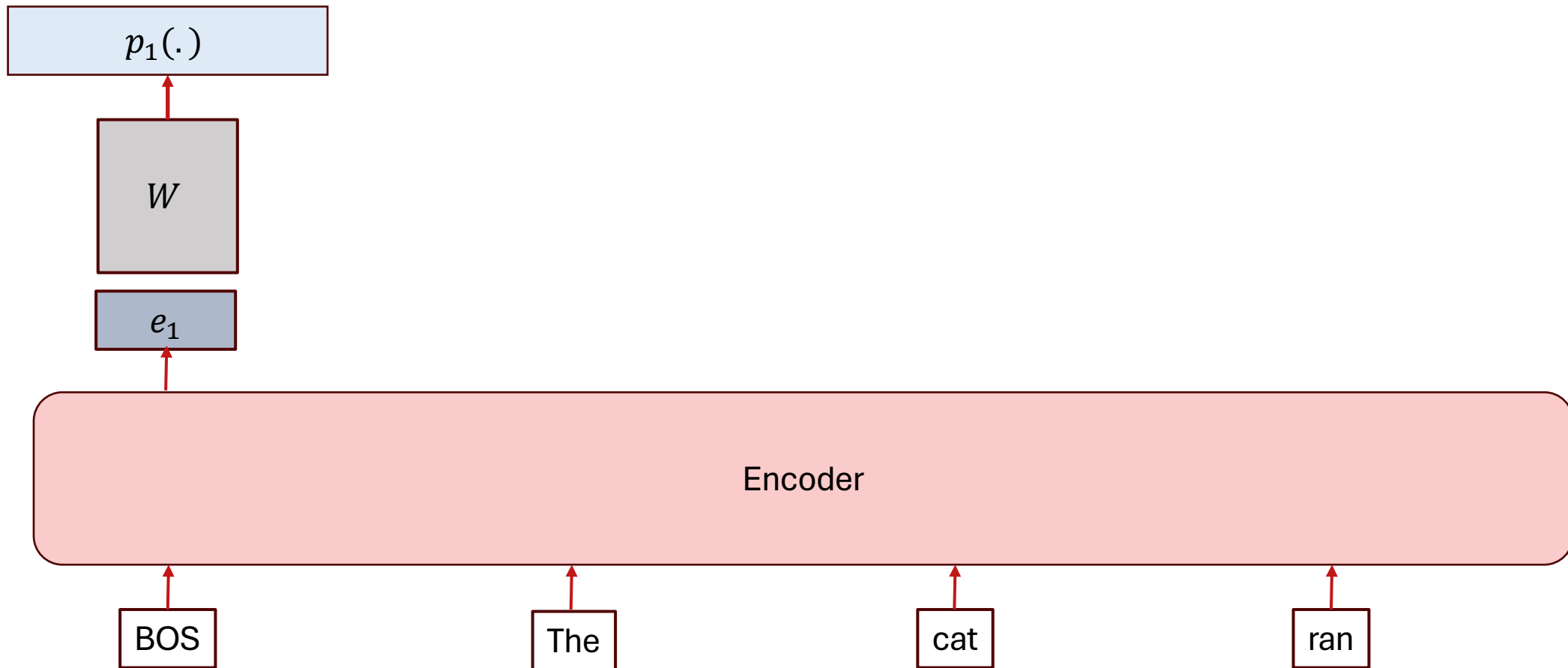
Sampling a sentence from an autoregressive model

- 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, \dots, t_j
$$t_{j+1} \sim p_{j+1}(\cdot | t_1, \dots, t_j)$$

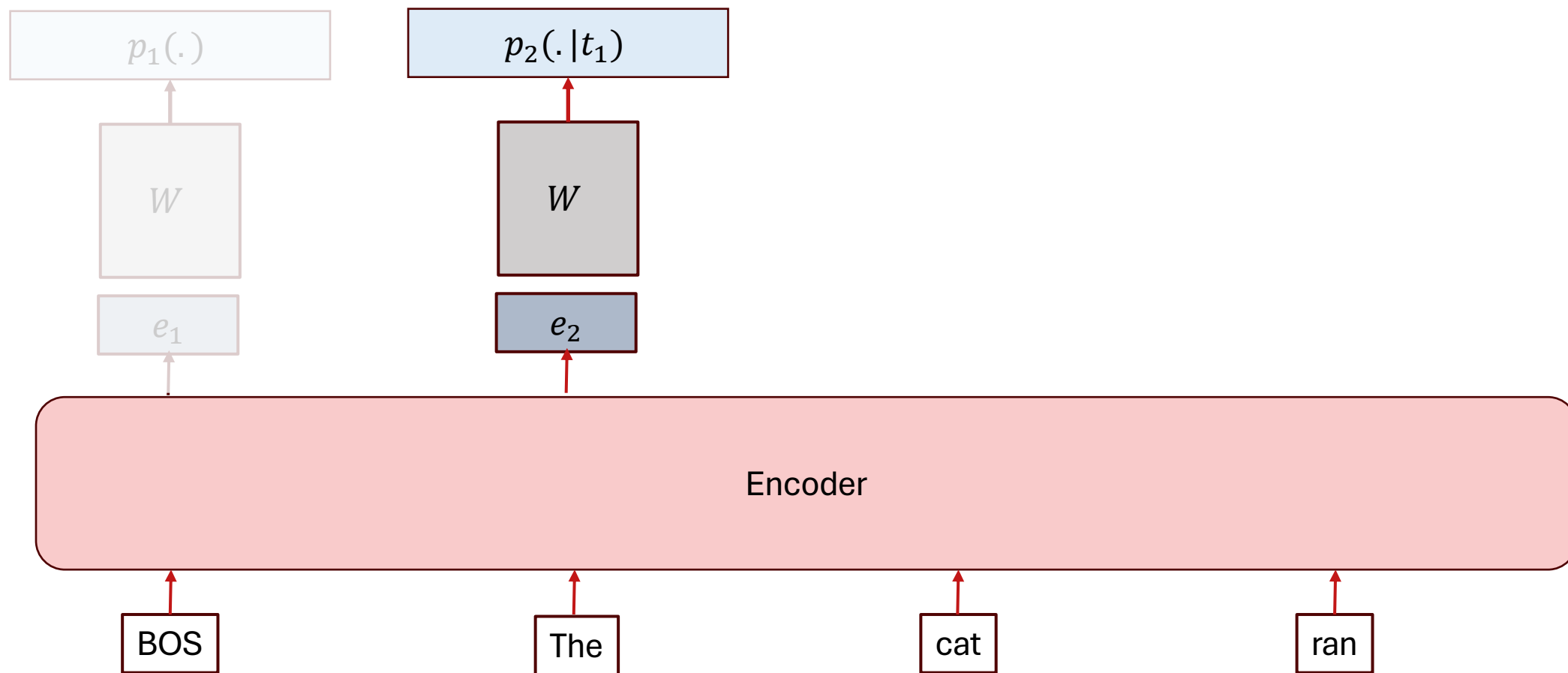
Since p_j are distributions over the vocabulary, they are easy to sample from !!



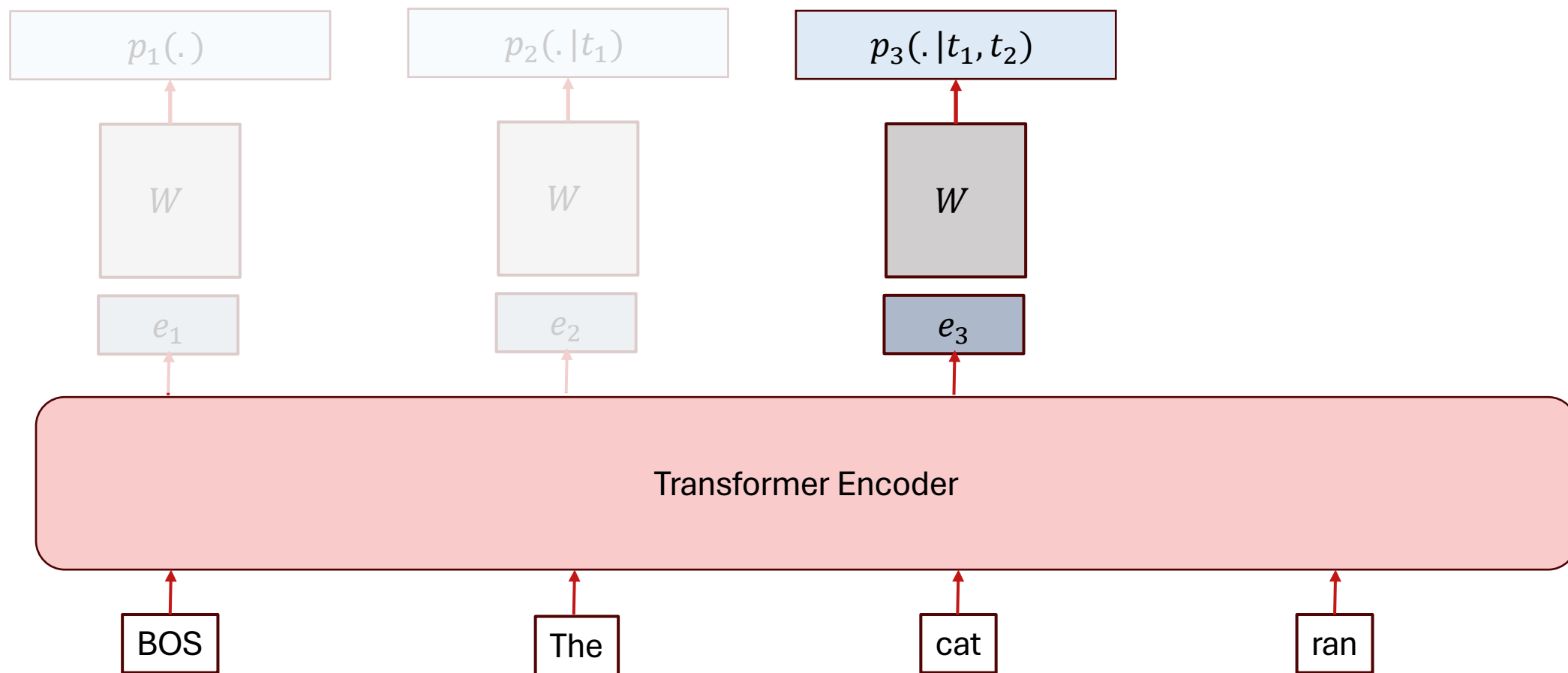
What do p_i s look like?



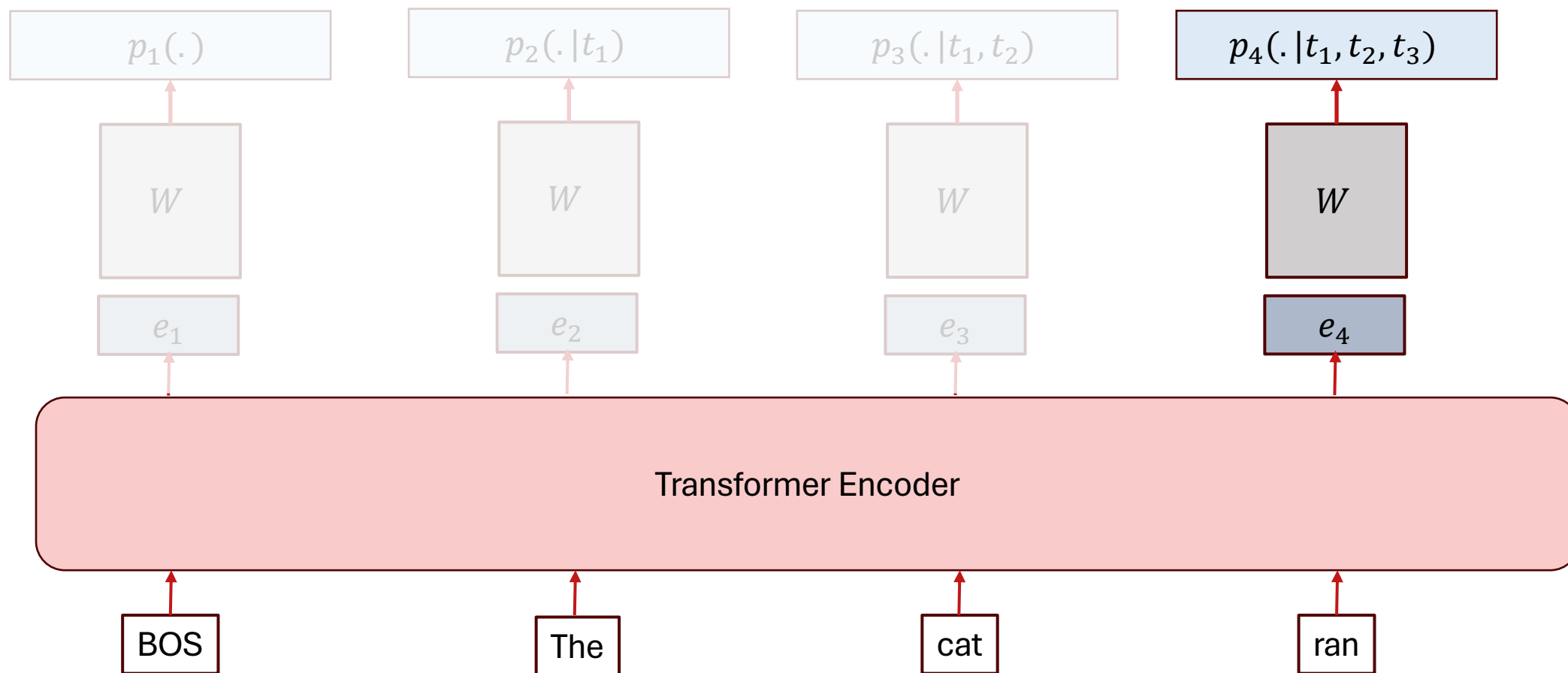
What do p_i s look like?



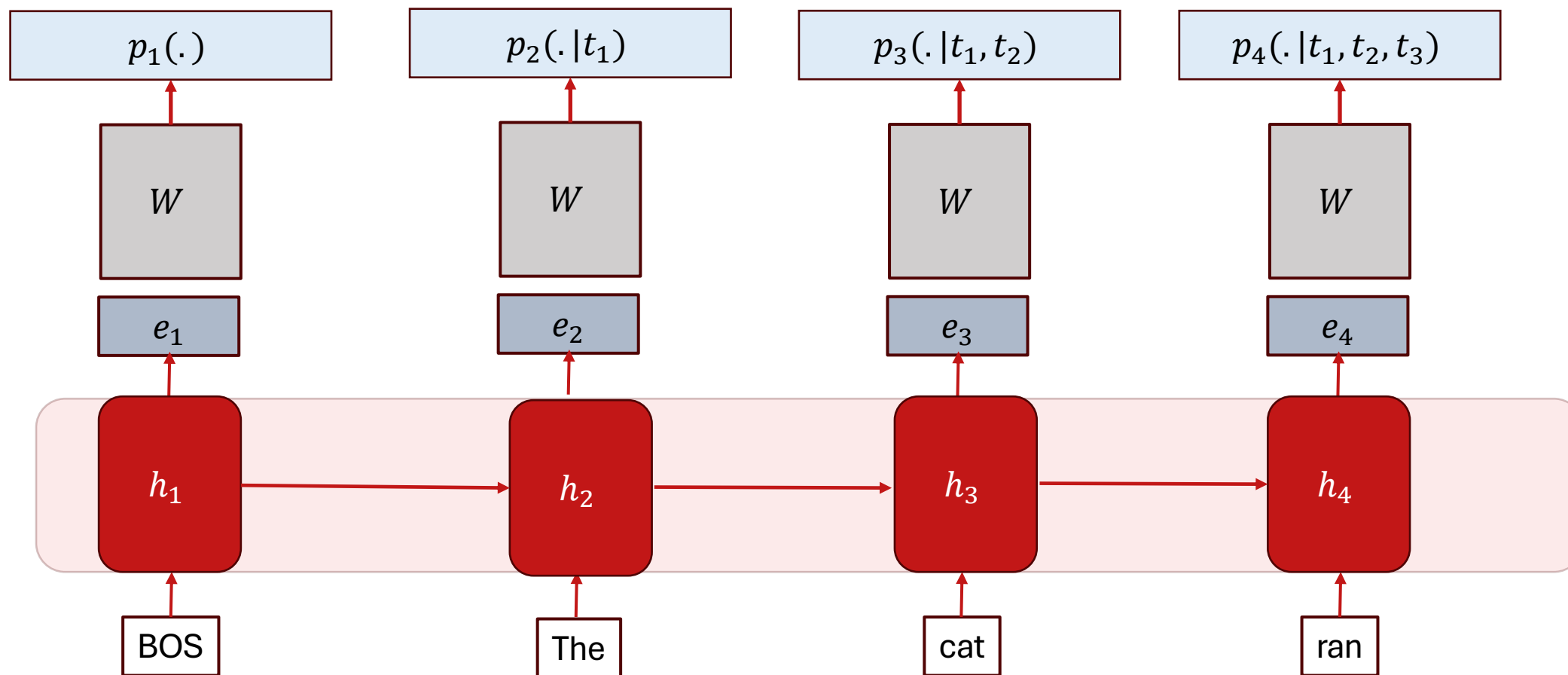
What do p_i s look like?



What do p_i s look like?



What do p_i s look like? – RNN encoder

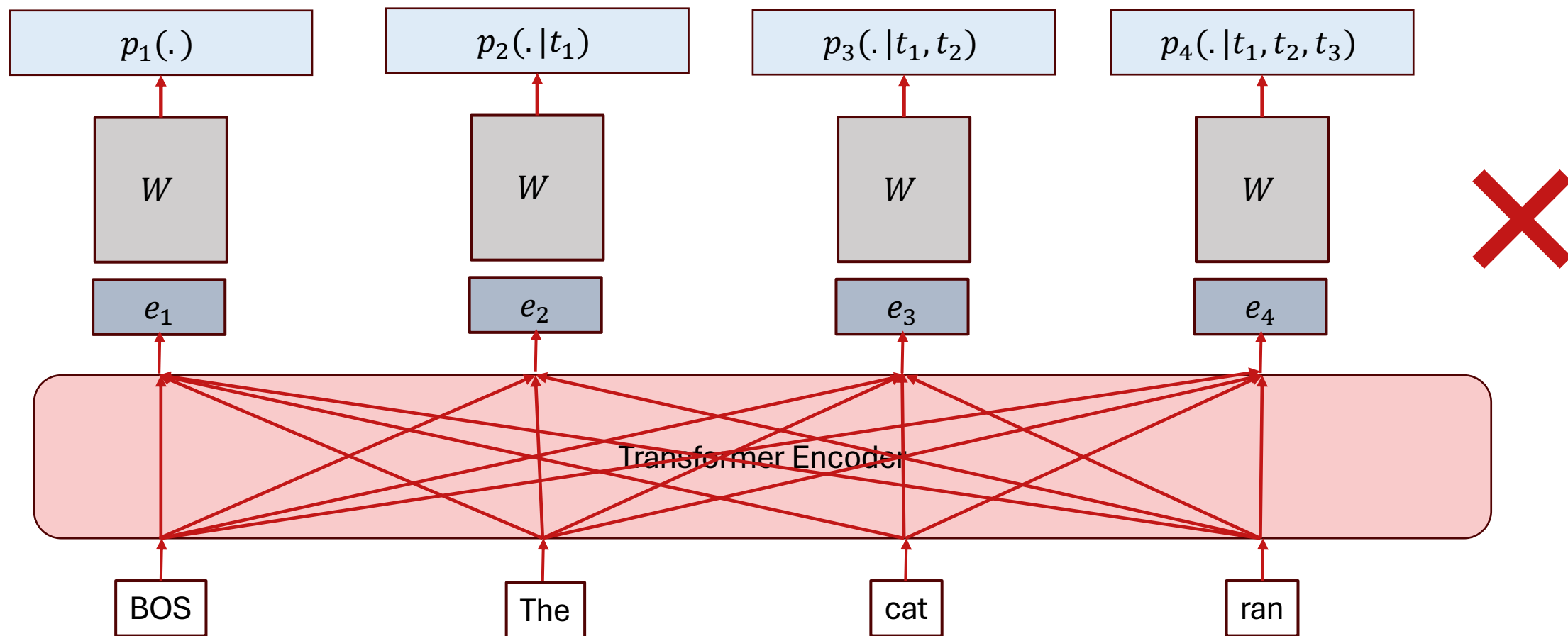


Problems with RNN Encoder

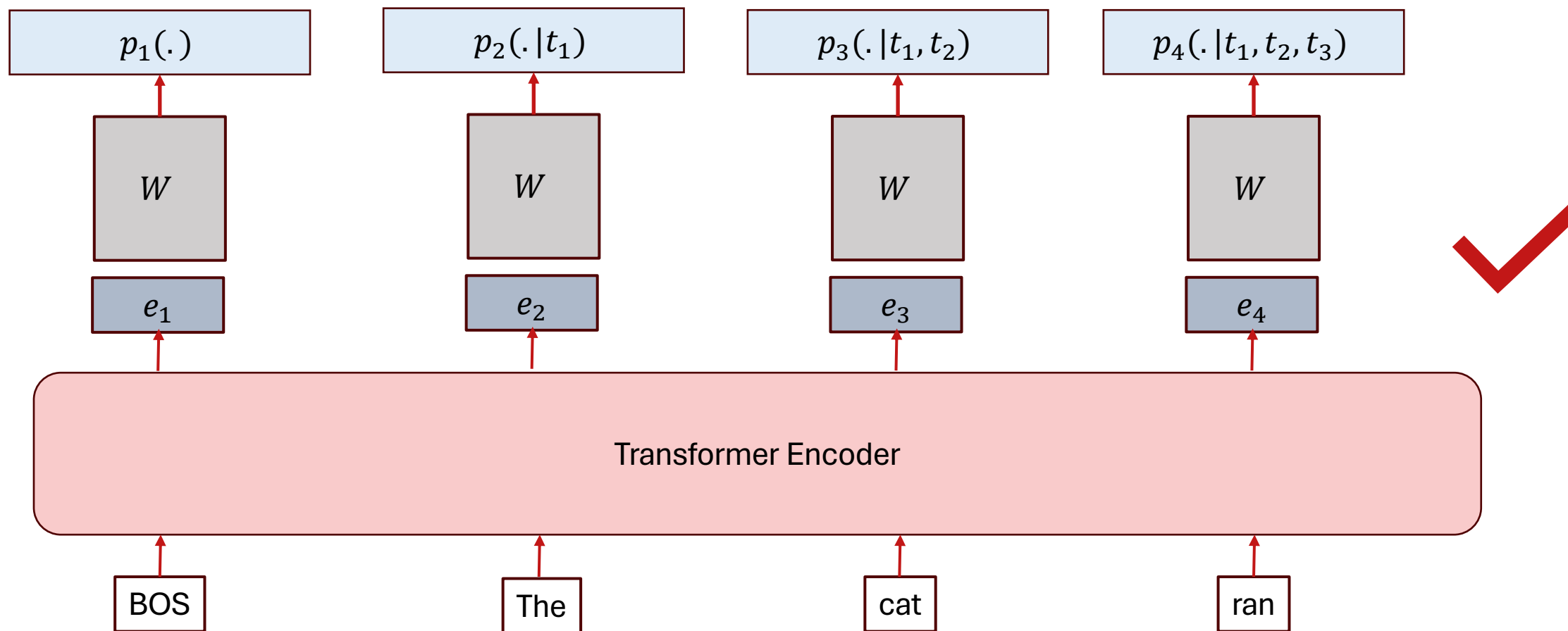
- The hidden state at $(j + 1)^{th}$ step depends on the hidden step at j^{th} step.
 - $h_{j+1} =$
 - Time \propto
- How to get rid of sequential computation ?



What do p_i s look like? – Full attention mask



What do p_i s 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, \dots, s_n

- Task

- Find the parameter θ^* that most likely resulted in the observed data, that is

$$\begin{aligned} & \arg \max_{\theta} p_\theta(s_1, \dots, s_n) \\ &= \arg \max_{\theta} \log p_\theta(s_1, \dots, s_n) \\ &= \arg \max_{\theta} \sum_{i=1}^n \log p_\theta(s_i) \end{aligned}$$



MLE estimation from sentence data

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

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

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

- Maximum Likelihood estimation



Pre-Training

Pre-Training Considerations

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



Data curation & filtering

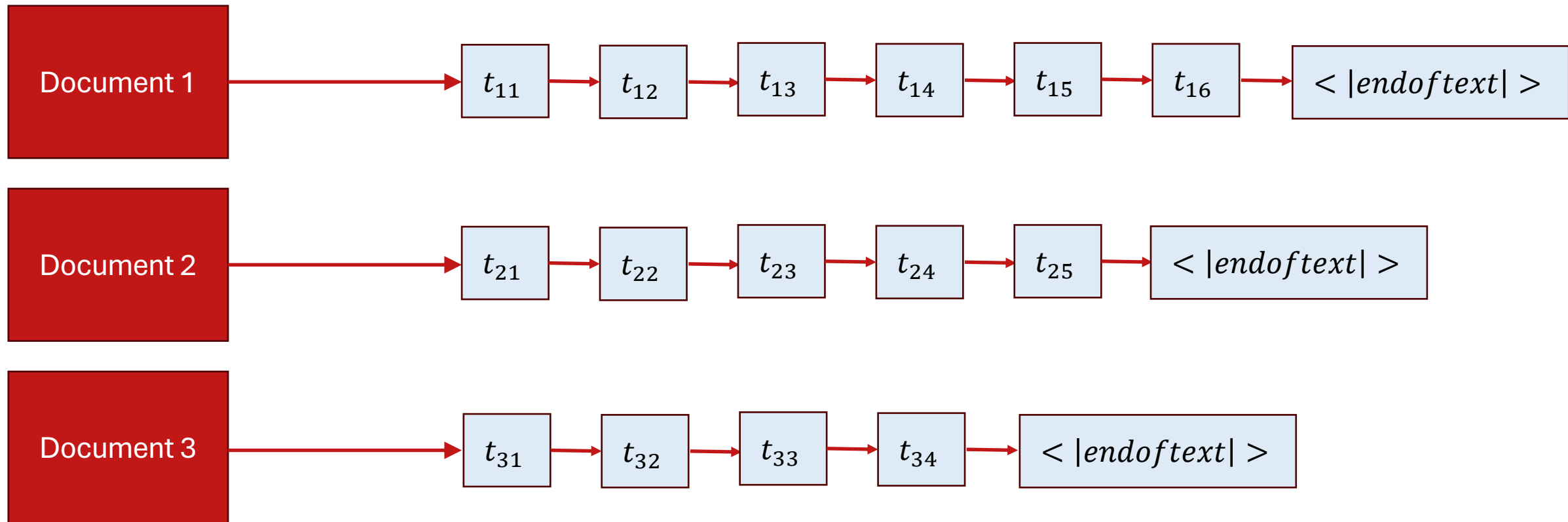
- Scrape the web for HTML/text documents
- Parse the HTML web page using HTML parsers
 - Removal of html tags
- Deduplication of web pages
- Remove non-informative documents – logs & error messages
- Remove outlier documents – documents with weird token distribution
- **Optional:** Train a classifier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

Credit: Llama 3.1 paper by Meta

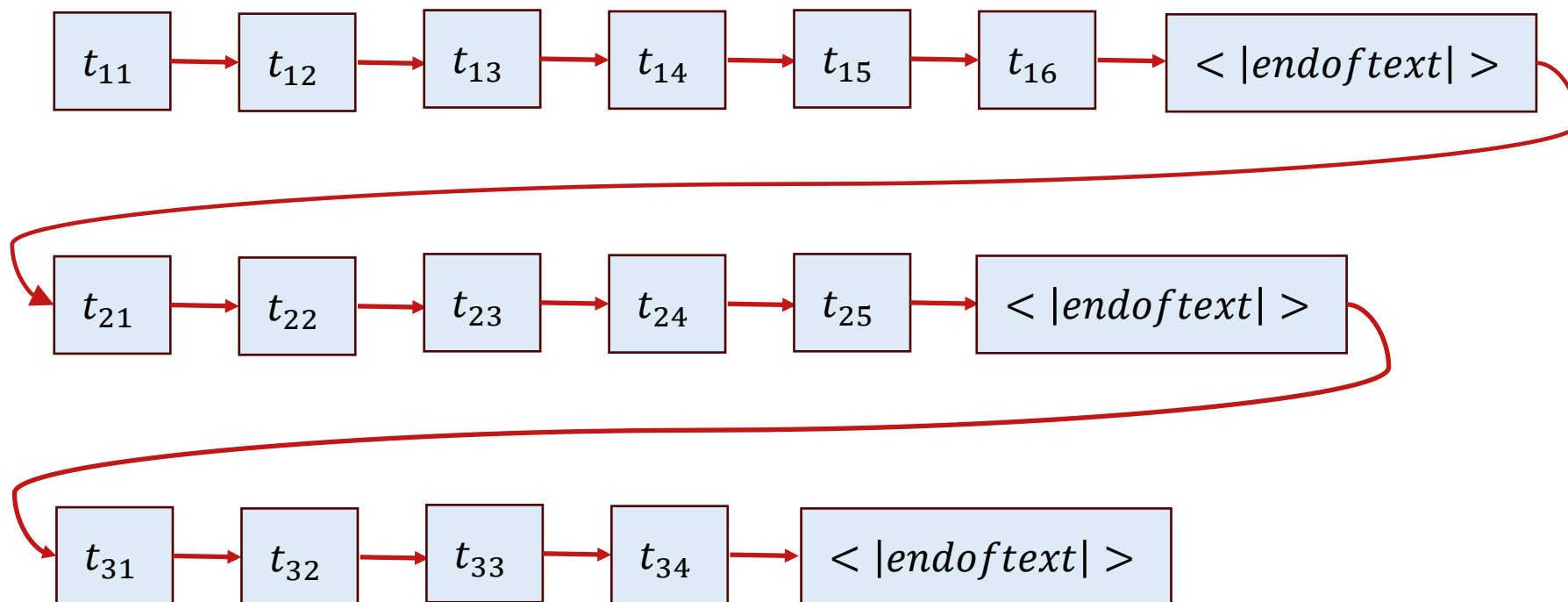


Tokenization & Data stream

Tokenize



Tokenization & Data stream



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**”.

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

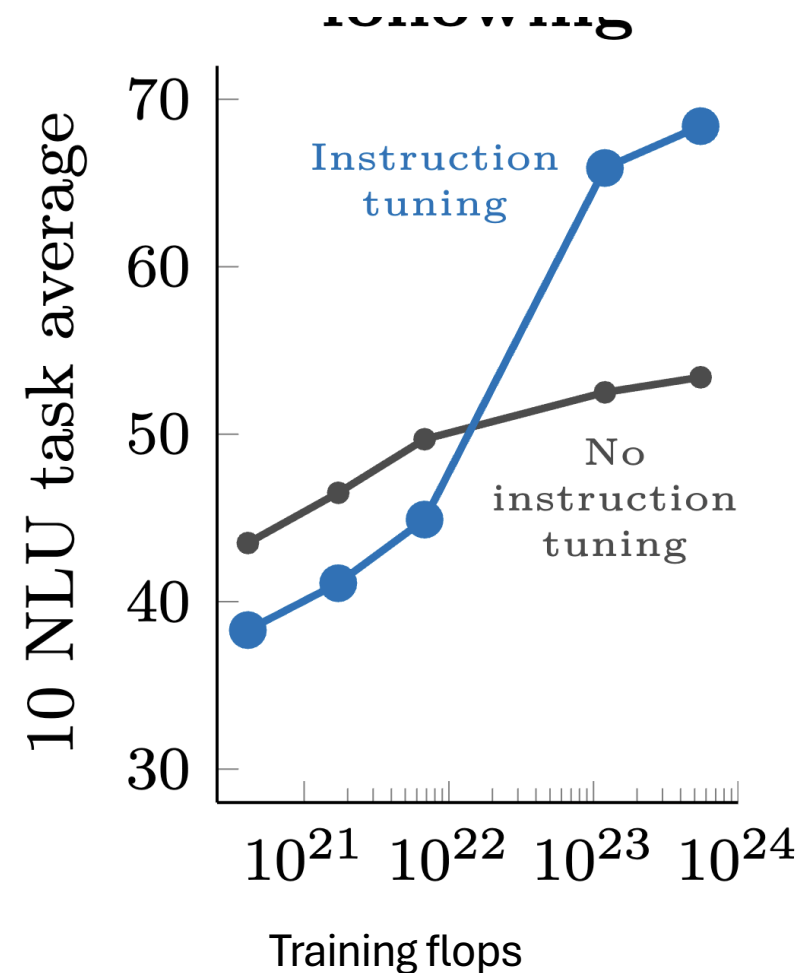
Model: Llama-3-8B

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



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

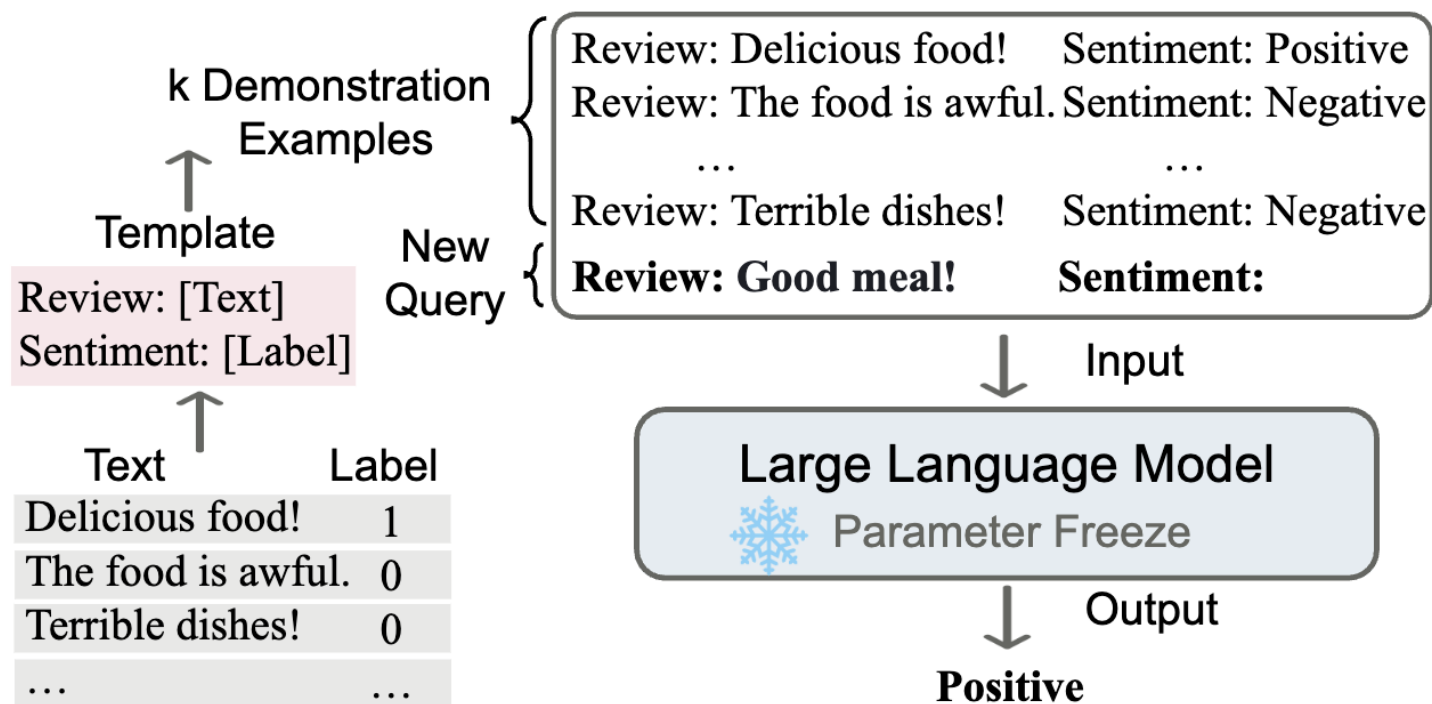


Emergence of In-context learning



What is in-context learning?

- Learning from a few examples within the context

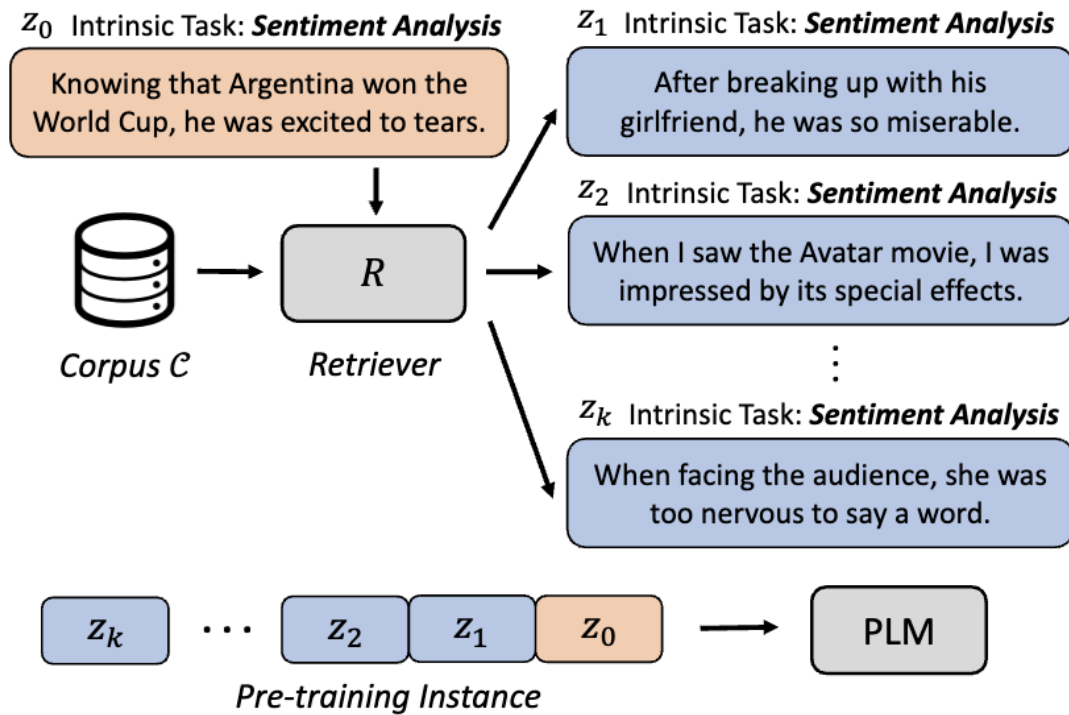


Credit: A Survey on In-context Learning



Boosting In-context learning - during pre-training

- Reorganize pretraining data to group similar examples together

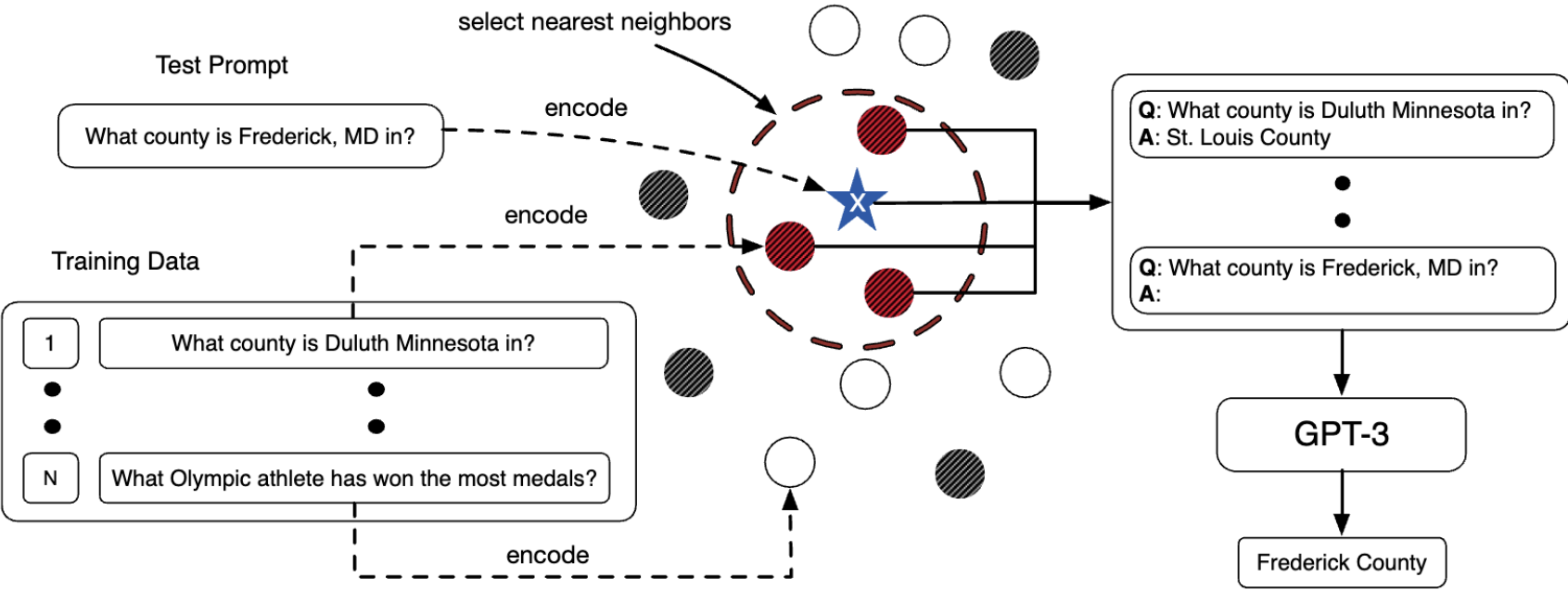


Credit: Pre-Training to Learn in Context



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

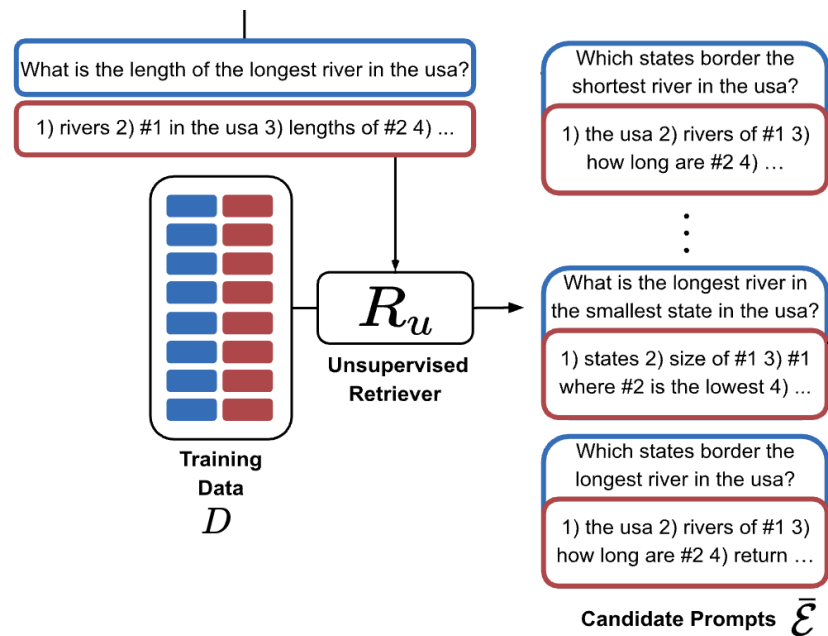
- Embed the inputs using a sentence embedder such as Roberta.
- Similar examples are those whose inputs maximize the cosine similarity.
- Given query q , the score of an example (x, y) is given by

$$Score(x) = \left\langle \frac{e(x)}{\|e(x)\|}, \frac{e(q)}{\|e(q)\|} \right\rangle$$

- The top-k examples are selected.
-
- Question: How can you make this set diverse?



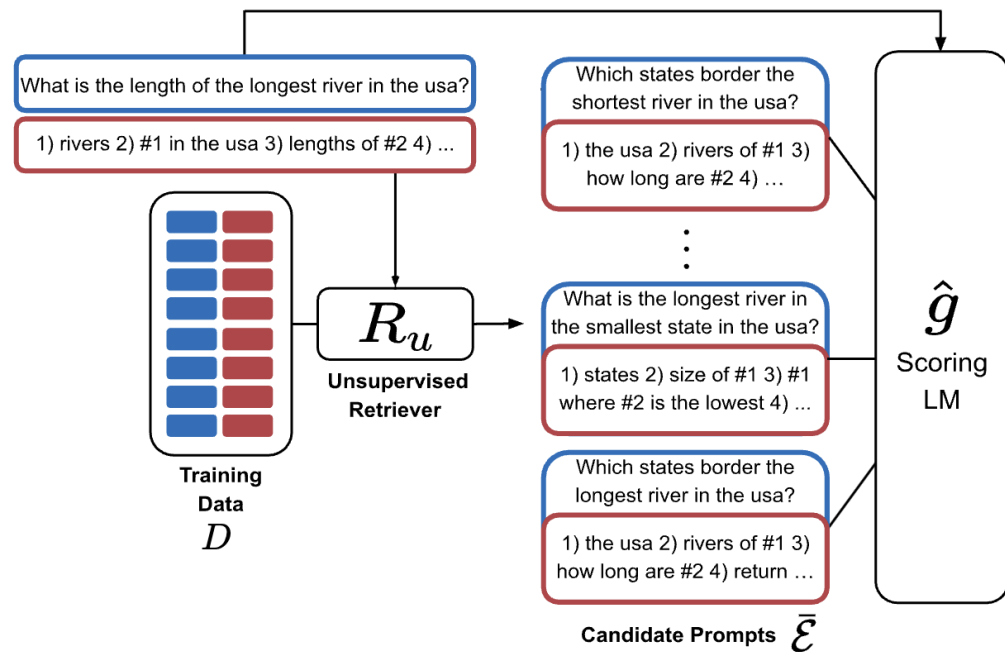
Contrastive learning of the similarity metric



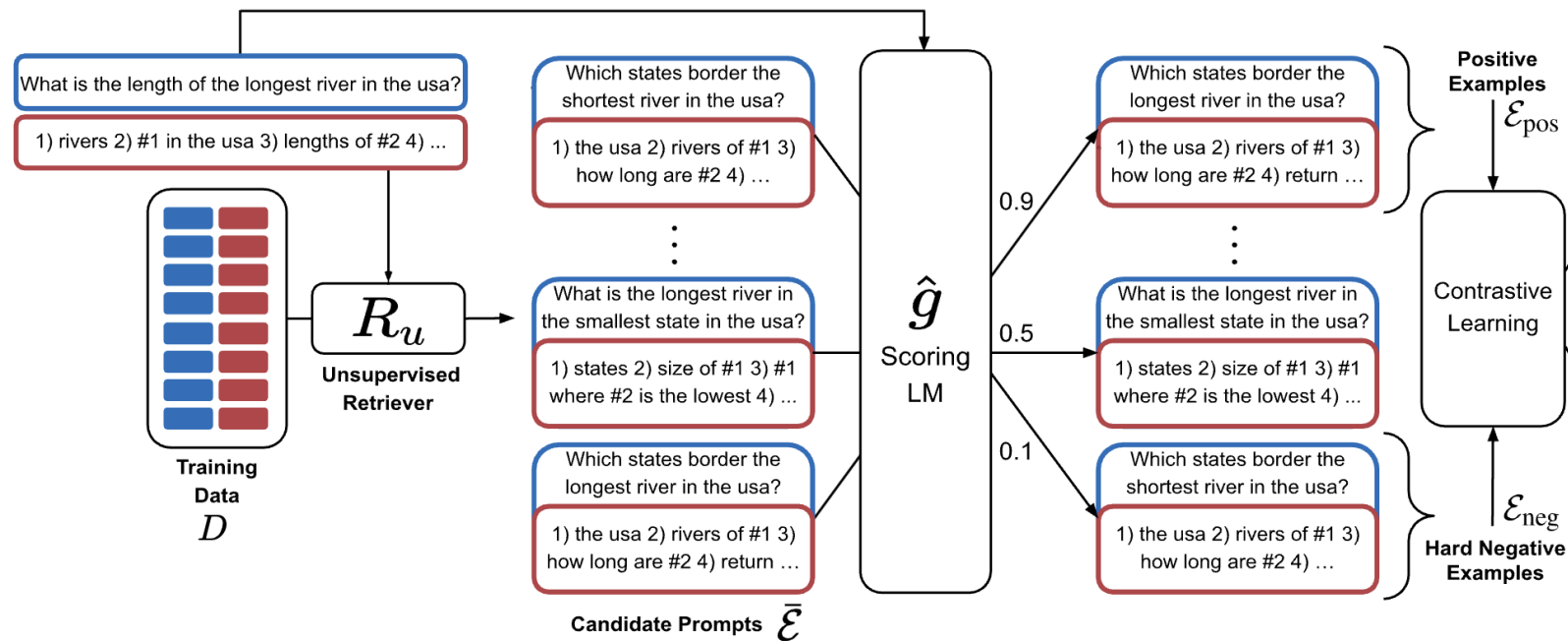
Credit: Learning To Retrieve Prompts for In-Context Learning



Contrastive learning of the similarity metric



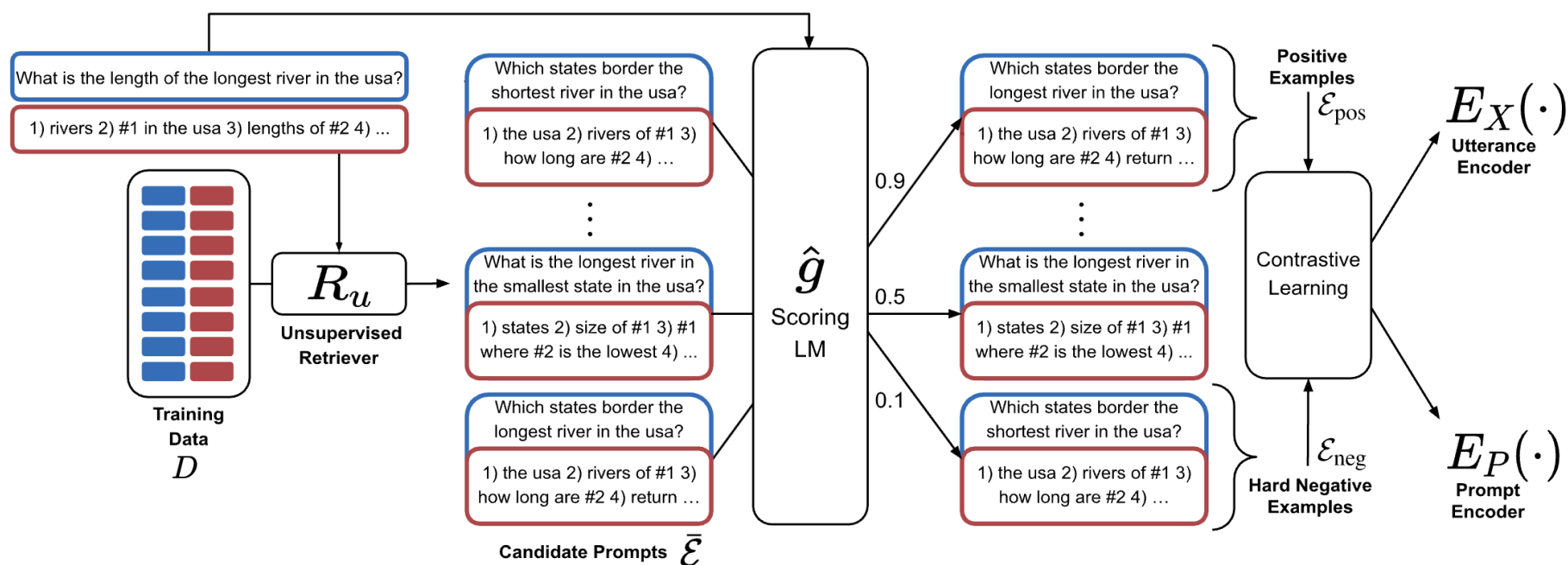
Contrastive learning of the similarity metric



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

$$\begin{array}{ll} x_1: (0.1, 0.3, -0.7), & y_1: -1.2 \\ x_2: (0.2, 0.4, -0.6), & y_2: -2.4 \\ \dots & \\ x_n: (0.4, 0.7, -0.3), & y_n: -1.6 \end{array}$$

Assumption: Linear model

$$y = \langle \theta, x \rangle$$

Loss $R(\theta) =$

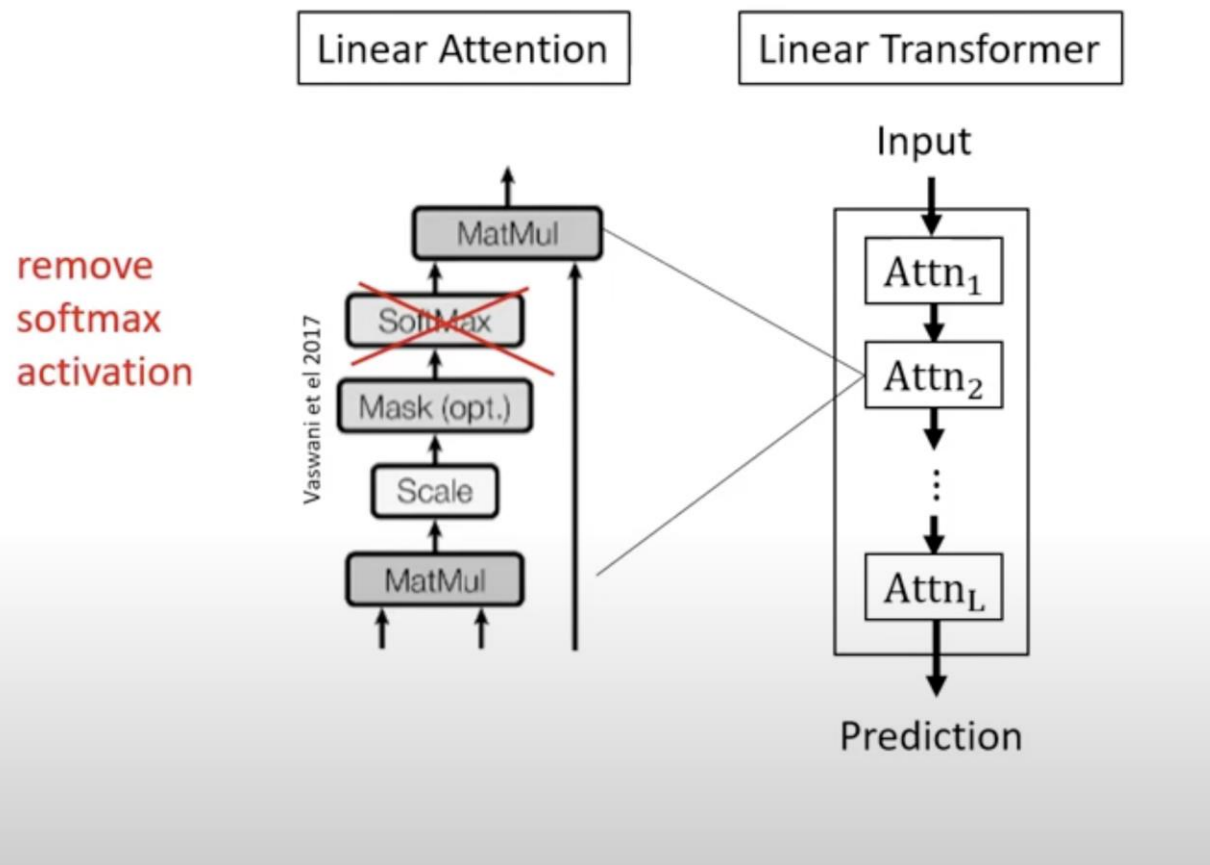
Gradient Descent

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

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



Linear transformers



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

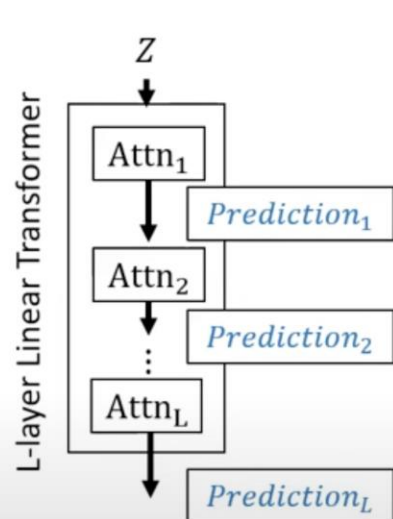


Transformers learn in-context by gradient descent

Input: $Z = \begin{bmatrix} x^{(1)} & x^{(2)} & \dots & x^{(n)} & x^{(n+1)} \\ y^{(1)} & y^{(2)} & \dots & y^{(n)} & 0 \end{bmatrix} \in \mathbb{R}^{(d+1) \times (n+1)}$

demonstrations
query

Goal: predict $y^{(n+1)}$



$$\theta_0^{GD} = 0$$

$$\theta_1^{GD} = \theta_0^{GD} - \delta_1 \nabla R(\theta_0^{GD})$$

$$\theta_2^{GD} = \theta_1^{GD} - \delta_2 \nabla R(\theta_1^{GD})$$

⋮

$$\theta_L^{GD} = \theta_{L-1}^{GD} - \delta_L \nabla R(\theta_{L-1}^{GD})$$

Gradient Descent on $R(\theta)$

Theorem 3 [ACDS 23]

Assume $x^{(i)} \sim \mathcal{N}(0, I)$, $\theta^* \sim \mathcal{N}(0, I)$. There **exist stationary points** of the training objective, such that the **Transformer prediction at layer k** is the same as k steps of gradient descent on $R(\theta)$, i.e.

$$\text{Prediction}_k = \langle \theta_k^{GD}, x^{(n+1)} \rangle,$$

for $k = 1 \dots L$

Training objective:

$$\mathbb{E} \left[\left(\text{Prediction}_L - y^{(n+1)} \right)^2 \right]$$

(True label)

$\hat{\theta}$ minimizes the Empirical Loss

$$R(\theta) = \sum_{i=1}^n (y^{(i)} - \theta^\top x^{(i)})^2$$

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.

