CS 489/698: Introduction to Natural Language Processing

Lecture 6.2: Language Modeling II

Instructor: Freda Shi

fhs@uwaterloo.ca

February 4th, 2026

UNIVERSITY OF WATERLOO

Outline: Neural Language Modeling

- Neural language modeling
- Probing neural language models

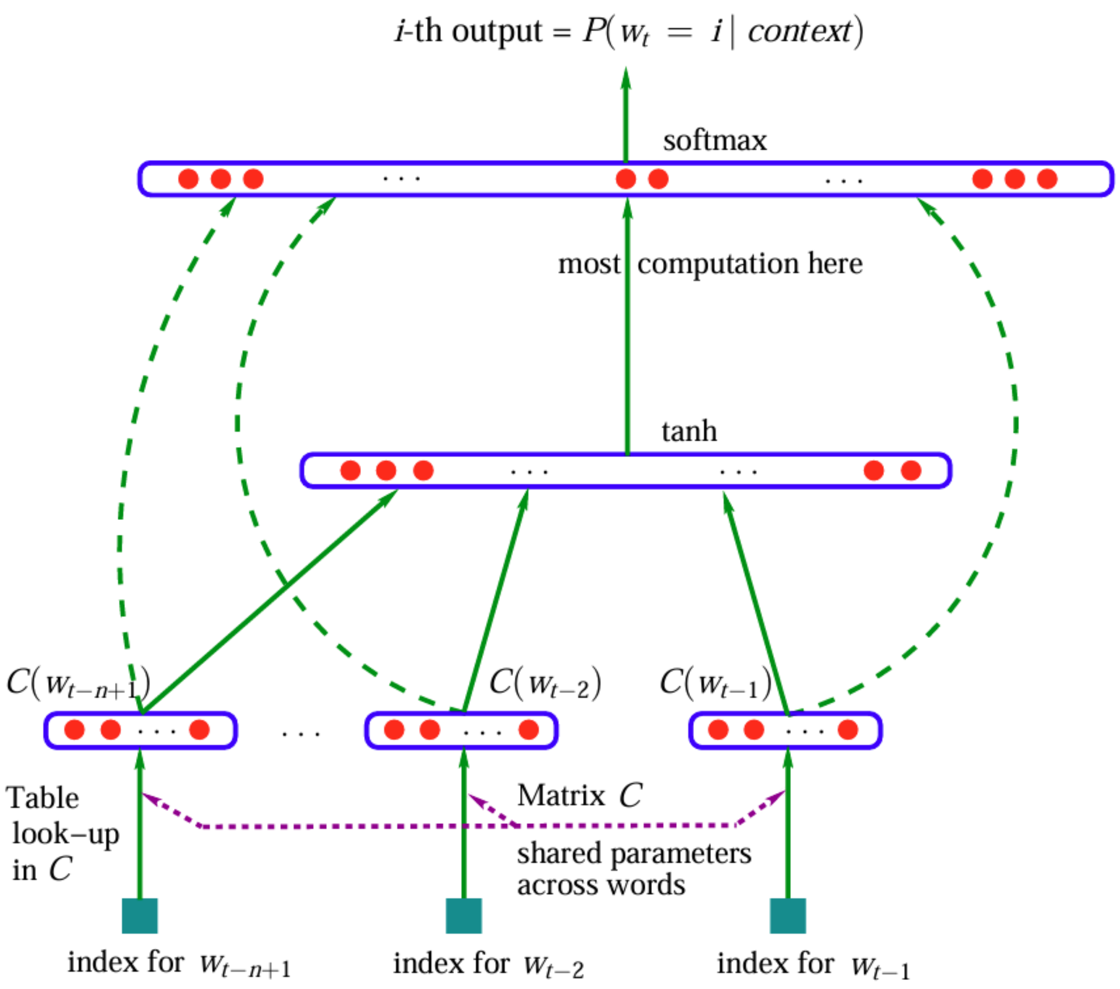
Recap: Language Modeling as Classification

$$P(\mathbf{x}_{1:n}) = P(x_1, x_2, \ldots, x_n) = P(\texttt{</s>} \mid x_1, \ldots, x_n) \prod_{i=1}^{n} \boxed{P(x_i \mid x_1, x_2, \ldots, x_{i-1})}$$

This is just a probabilistic classification problem!

We can use any tools from the previous lectures: linear model with features, neural networks, etc.

Neural N-Gram Language Models



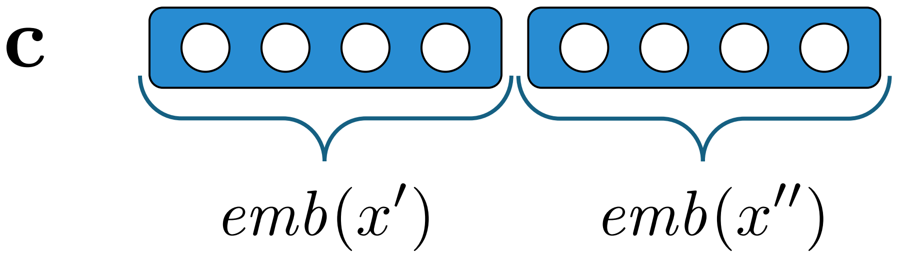
[Src: Bengio et al., 2003]

Neural Trigram Language Model

Given two previous words, compute probability distribution over possible next words

$$P(x \mid x', x'')$$

Input is concatenation of vectors (embeddings) of previous two words:

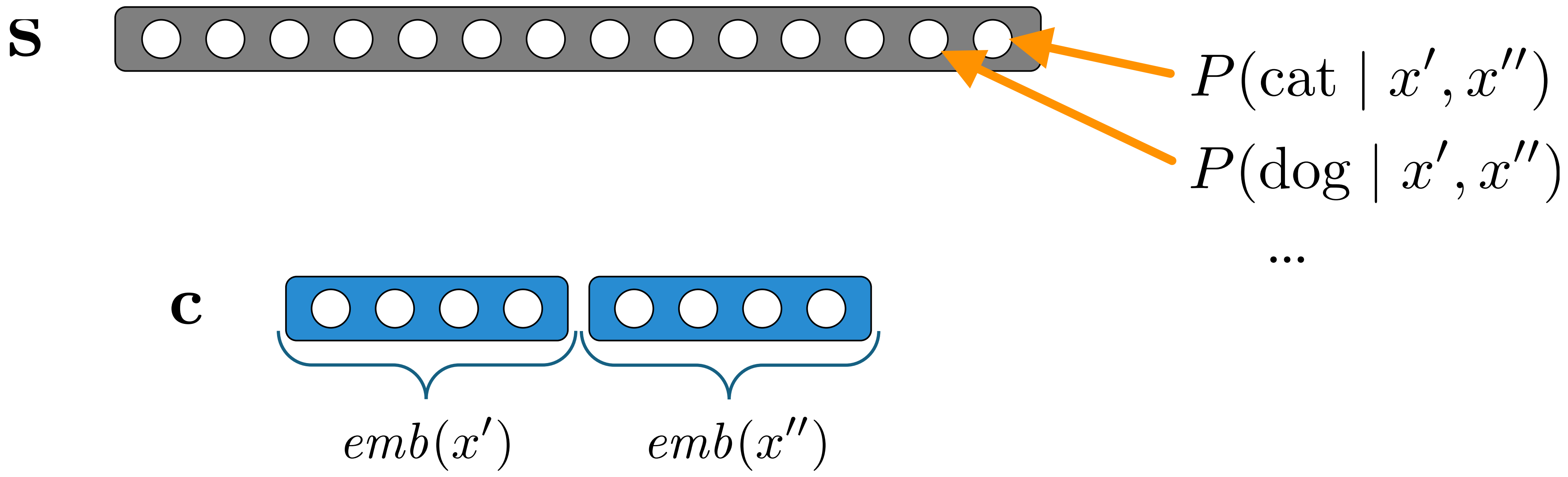


$$emb(\cdot) \in \mathbb{R}^d$$

$$\mathbf{c} = cat(emb(x'), emb(x''))$$

Neural Trigram Language Model

Output is a vector $\mathbf{s}$ containing probabilities of all possible next words:

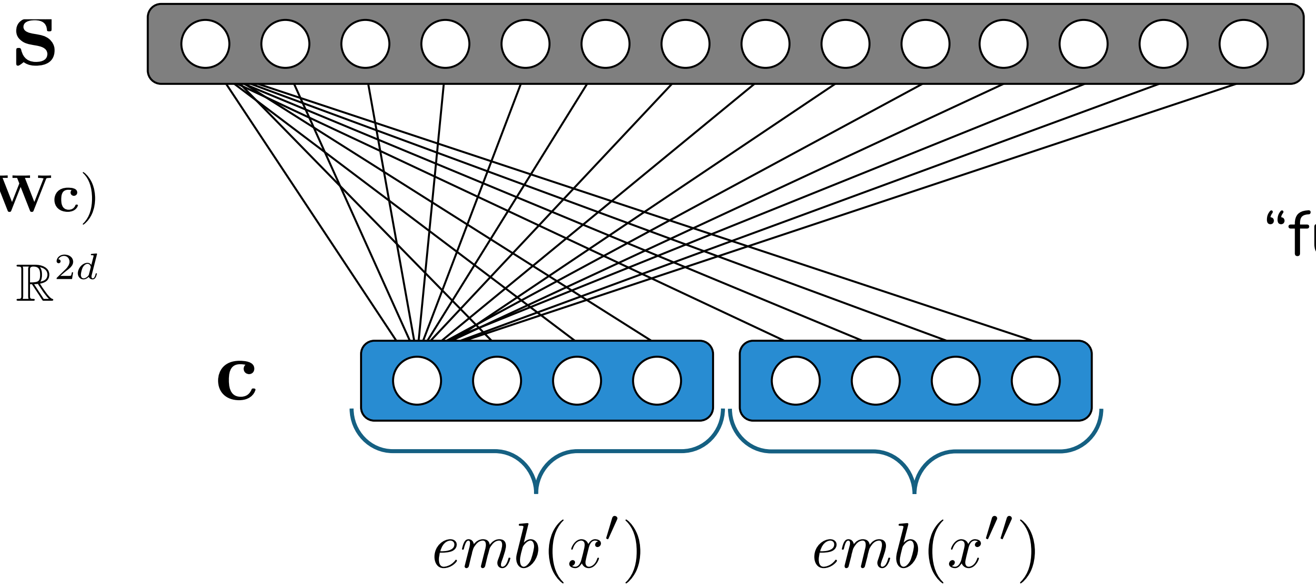


Neural Trigram Language Model

To get $\mathbf{S}$, do matrix multiplication of parameter matrix $\mathbf{W}$ and input, then "softmax" transformation

$$\mathbf{s} = \mathrm{softmax}(\mathbf{W}\mathbf{c})$$

$$\mathbf{W} \in \mathbb{R}^{|V| \times 2d} \quad \mathbf{c} \in \mathbb{R}^{2d}$$



"fully-connected layer"

Neural Trigram Language Model

$$P_\Theta(x_t \mid x_1, \ldots, x_{t-1}) = s_{x_t}$$

$$\text{loss } L = -\sum_i \log P_\Theta(\mathbf{x}^{(i)}) = -\sum_i \sum_j \log P_\Theta(x_j^{(i)} \mid x_1^{(i)}, \ldots, x_{j-1}^{(i)})$$

$\nabla_\Theta L$ via backpropagation.

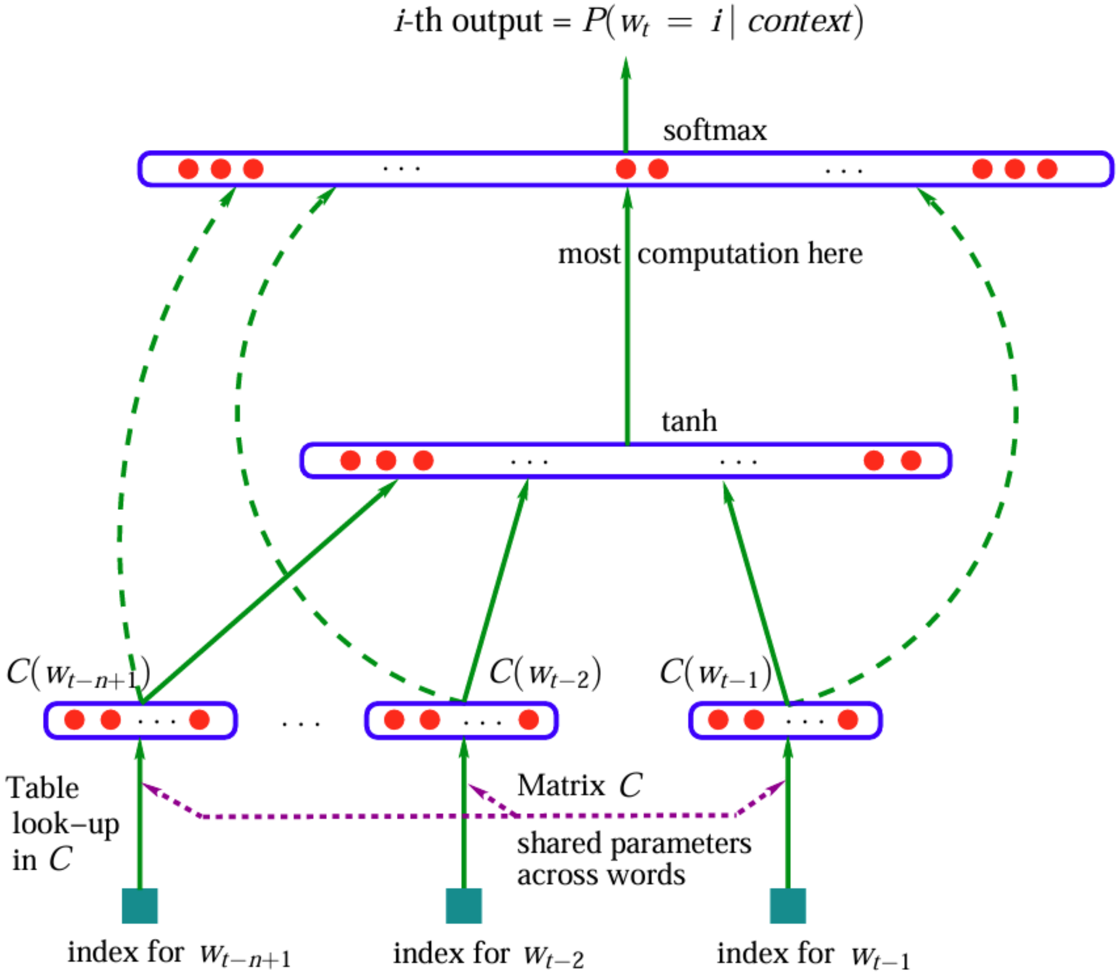
Trigram vs. Neural Trigram LMs $P(x \mid x', x'')$

- **Trigram language model**
  - separate parameters for every combination of x, x', x''
  - so, approx. $|V|^3$ parameters
  - # parameters is exponential in n-gram size
  - most parameters are zero
  - even with smoothing, many parameters can remain zero

- **Neural trigram language model**
  - only has $\mathcal{O}(d|V|)$ parameters
  - d can be chosen to scale # parameters up or down
  - # parameters linear in n-gram size
  - (almost) no parameters are zero
  - no explicit smoothing, though smoothing done implicitly via distributed representations

Removing N-Gram Constraints

$$P(x_i \mid x_1, x_2, \ldots, x_{i-1})$$



[Src: Bengio et al., 2003]

$$x_i \xrightarrow{\text{look-up in } \mathbf{U}} \mathbf{w}_i \in \mathbb{R}^d$$

$$x_1, x_2, \ldots, x_{i-1} \rightarrow \frac{1}{i-1} \sum_{j=1}^{i-1} \mathbf{w}_j \in \mathbb{R}^d$$

$$P(x_i = v \mid x_1, x_2, \ldots, x_{i-1})$$

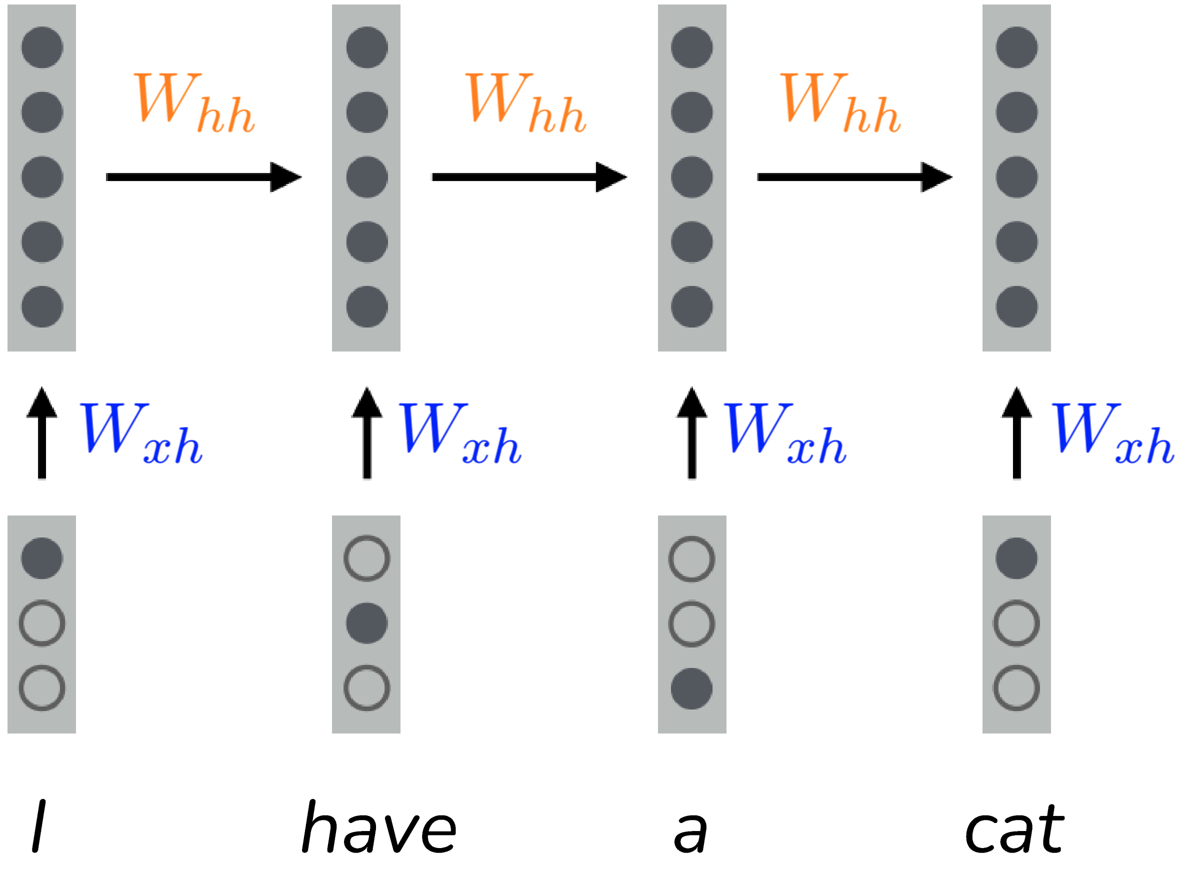
$$= \text{softmax}_v \mathbf{W} \left(\frac{1}{i-1} \sum_{j=1}^{i-1} \mathbf{w}_j \right)$$

$$\mathbf{W} \in \mathbb{R}^{|V| \times d}$$

RNN Language Models

Hidden state is a function of previous hidden state and current input.

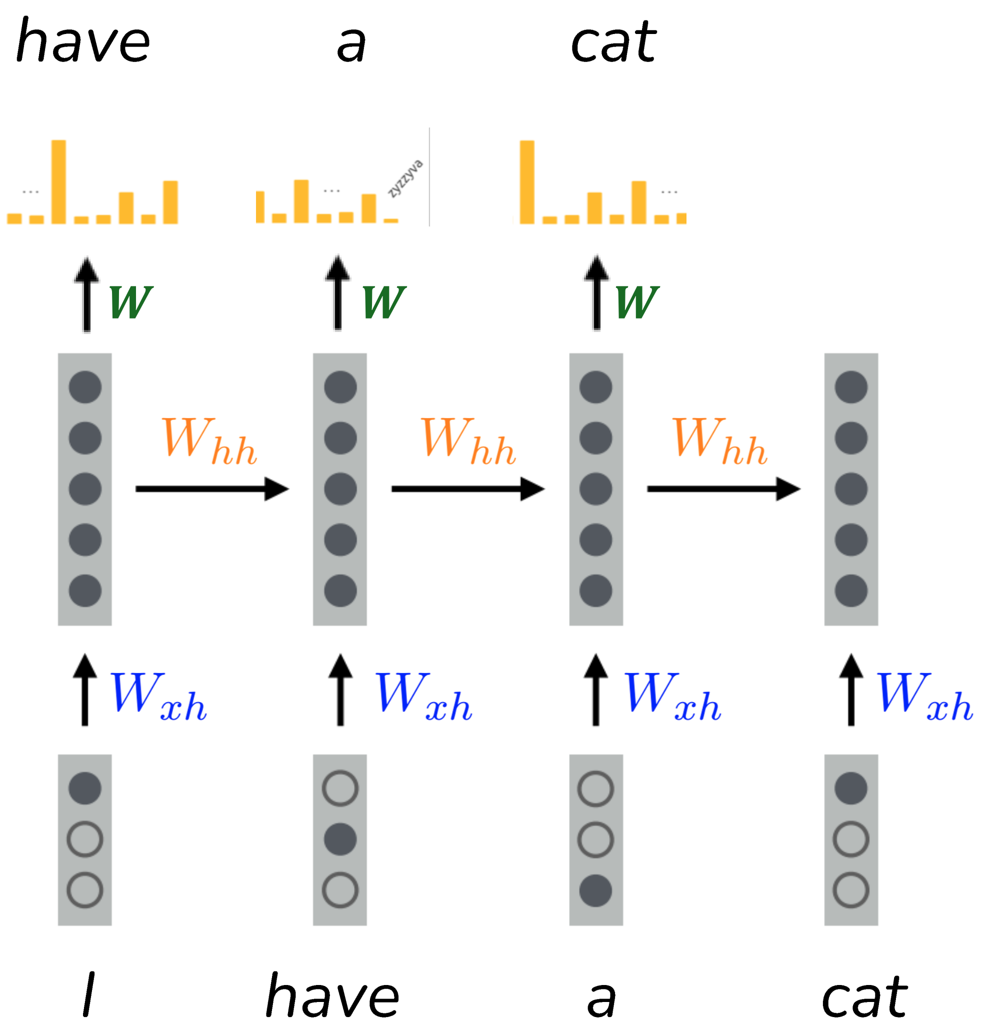
Same weights at each state.



$$\mathbf{h}_t = f(\mathbf{W}_{hh}\mathbf{h}_{t-1} + \mathbf{W}_{xh}\mathbf{x}_t)$$

$$P_\Theta(x_t \mid x_1, \ldots, x_{t-1}) = \text{softmax}_{x_t}\left(\mathbf{W}\mathbf{h}_{t-1}\right)$$

RNN Language Model



$$L_t = -\log P_\Theta(x_t \mid x_1, \ldots, x_{t-1})$$

$$\Theta = \{\mathbf{U}, \mathbf{W}_{xh}, \mathbf{W}_{hh}\}$$

$$\begin{aligned} L &= \sum_t L_t \\ &= \sum_t -\log P_\Theta(x_t \mid x_1, \ldots, x_{t-1}) \\ &= -\log P_\Theta(\mathbf{x}) \end{aligned}$$

$\nabla_\Theta L$ via backpropagation.

Transformer Language Models

A token "attends" to all previous tokens.

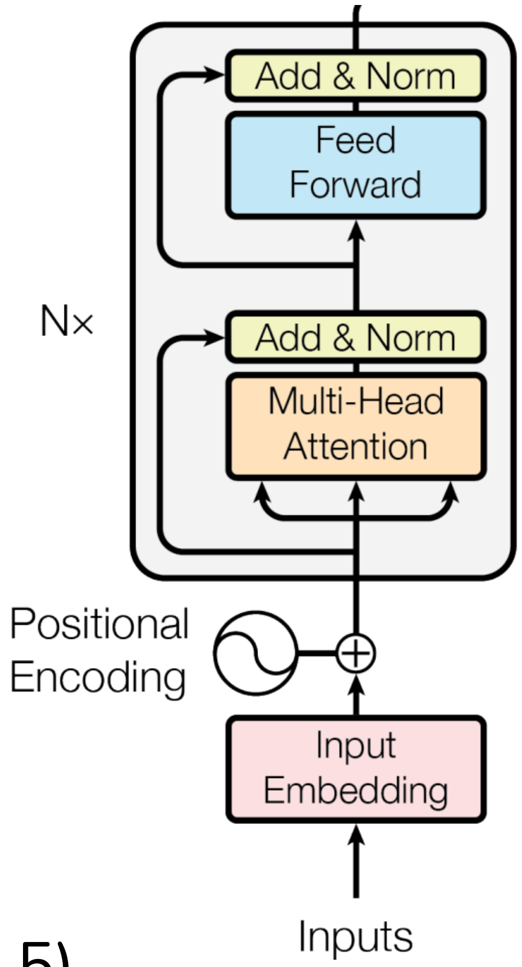
$$\mathbf{E} = (emb(w_1), \ldots, emb(w_k)) + \mathbf{P} \in \mathbb{R}^{d_1 \times k}$$

$$\mathbf{K} = \mathbf{W}_k \mathbf{E} \quad \mathbf{W}_k \in \mathbb{R}^{d_2 \times d_1}$$

$$\mathbf{Q} = \mathbf{W}_q \mathbf{E} \quad \mathbf{W}_q \in \mathbb{R}^{d_2 \times d_1}$$

$$\mathbf{V} = \mathbf{W}_v \mathbf{E} \quad \mathbf{W}_v \in \mathbb{R}^{d_3 \times d_1}$$

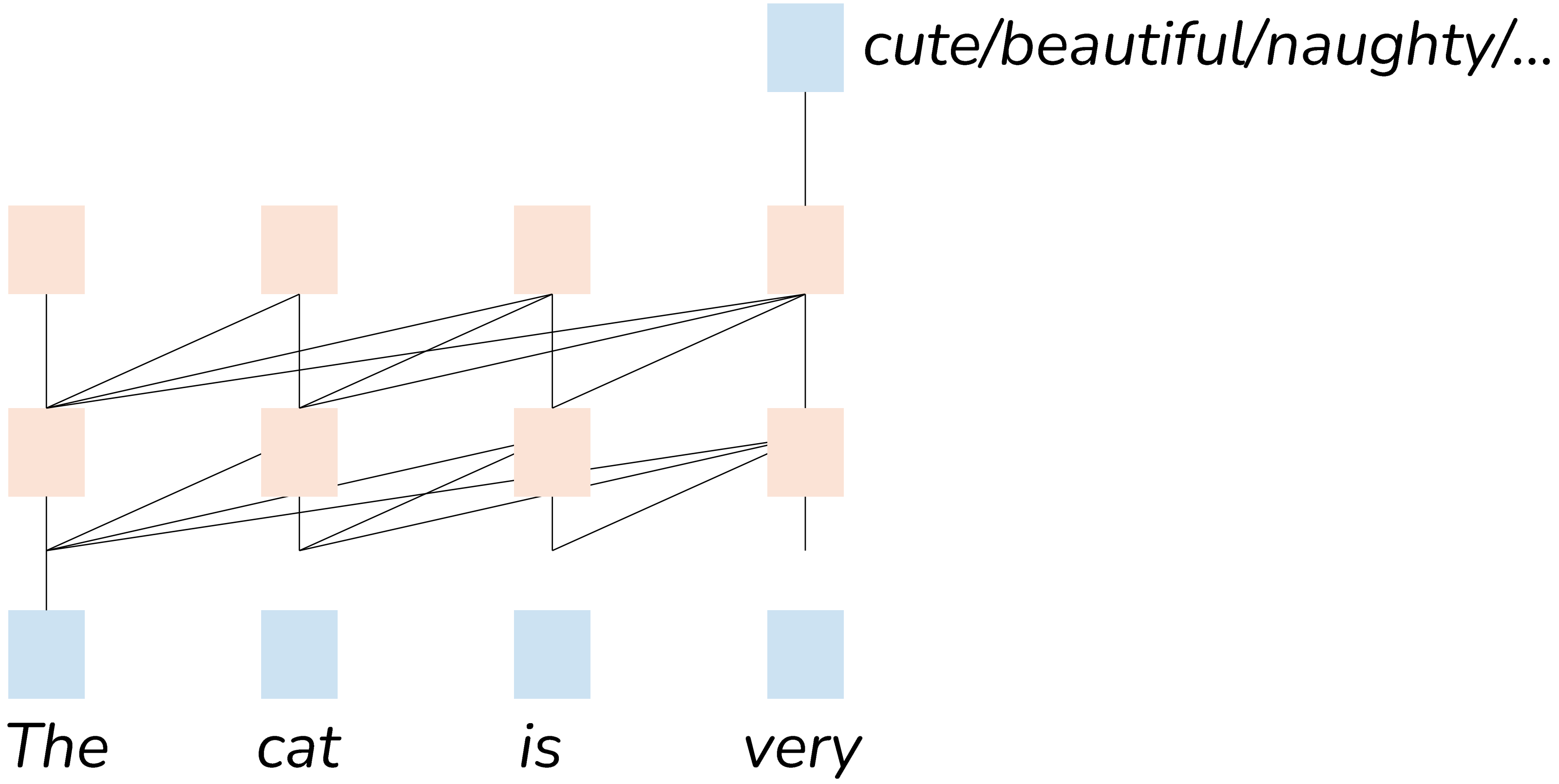
$$\tilde{\mathbf{E}} = \mathbf{V}\mathrm{softmax}\left(\frac{\mathbf{K}^\top \mathbf{Q}}{\sqrt{d_2}}\right) \in \mathbb{R}^{d_3 \times k}$$



Use $\tilde{\mathbf{E}}_{\cdot,k}$ as feature to predict the next token.

Note: feature is more complicated in real practice (see Lecture 5).

Language models encode knowledge about language

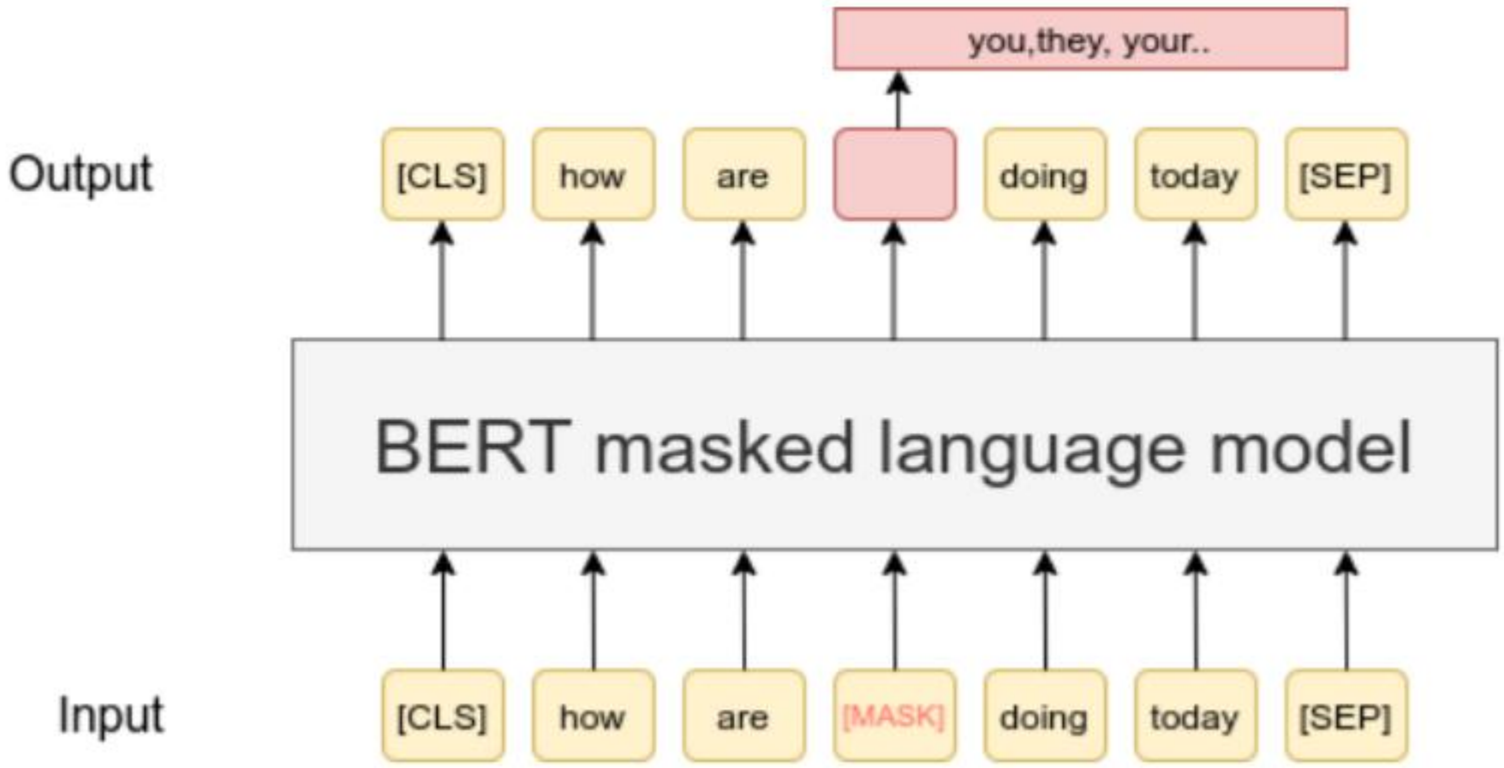


The pretraining-finetuning paradigm: Language modeling, as the pretraining task, helps encode knowledge.

The knowledge helps downstream tasks.

Masked Language Models

Motivation: learning useful representations of text.



[Src: https://www.sbert.net/examples/sentence_transformer/unsupervised_learning/MLM/README.html]

Mased Language Models

A token "attends" to all context tokens.

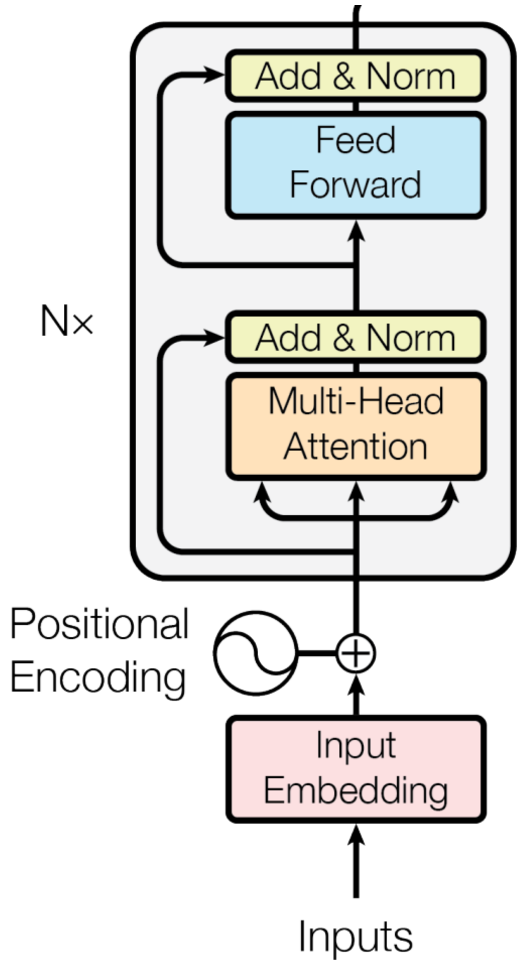
$$\mathbf{E} = (emb(w_1), \ldots, emb(w_k)) + \mathbf{P} \in \mathbb{R}^{d_1 \times k}$$

$$\mathbf{K} = \mathbf{W}_k\mathbf{E} \quad \mathbf{W}_k \in \mathbb{R}^{d_2 \times d_1}$$

$$\mathbf{Q} = \mathbf{W}_q\mathbf{E} \quad \mathbf{W}_q \in \mathbb{R}^{d_2 \times d_1}$$

$$\mathbf{V} = \mathbf{W}_v\mathbf{E} \quad \mathbf{W}_v \in \mathbb{R}^{d_3 \times d_1}$$

$$\tilde{\mathbf{E}} = \mathbf{V}\mathrm{softmax}\left(\frac{\mathbf{K}^\top\mathbf{Q}}{\sqrt{d_2}}\right) \in \mathbb{R}^{d_3 \times k}$$



Replace token at position i with a placeholder [MASK].

Use $\tilde{\mathbf{E}}_{\cdot,i}$ as feature to predict token at position i.

Some Important Details of LMs

- The importance of the held-out data.
- AdamW (Kingma & Ba, 2015; Loshchilov & Hutter) has become the go-to optimizer instead of vanilla gradient descent.
- 1×10^{-4} could be a default learning rate for AdamW.
- Monitor your training loss through time (by printing it out or using loggers like weights & biases; https://wandb.ai/site/).

Check out the HuggingFace Tutorial on training language models:
https://huggingface.co/learn/llm-course/en/chapter7/6

Probing

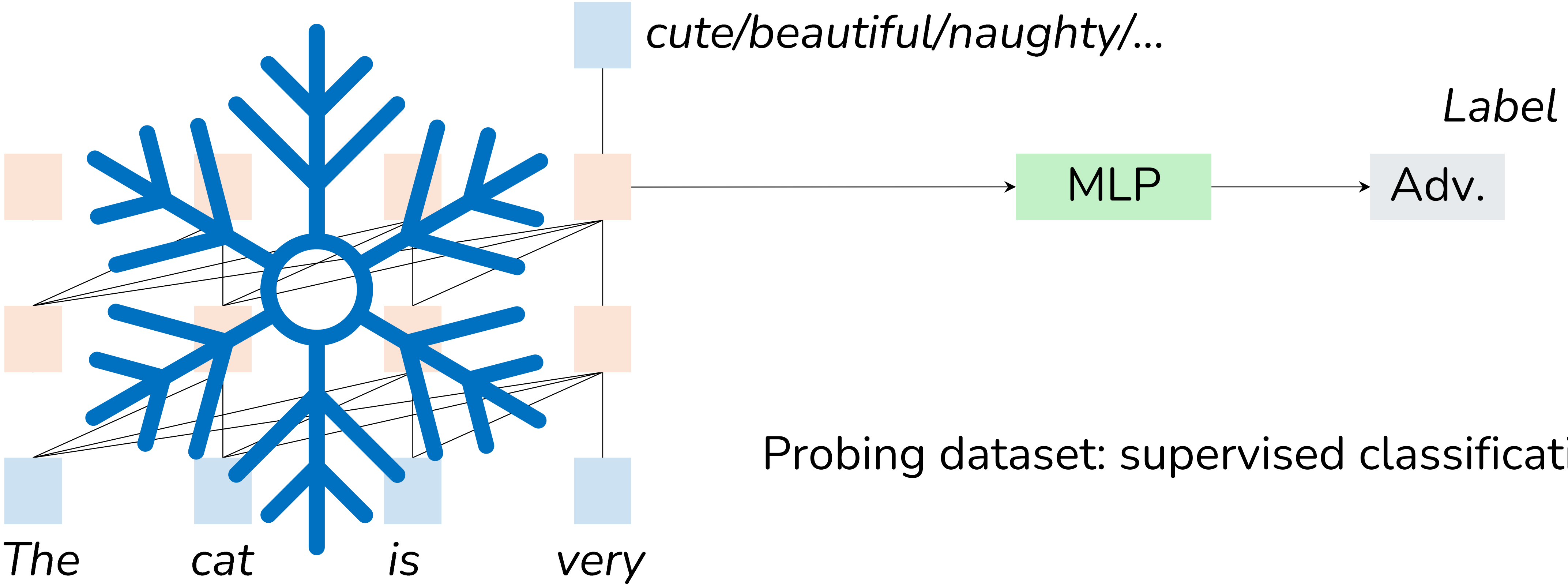
What is encoded in a trained neural language model?

Empirical answer: linguistic probe (Ettinger et al., 2016).

Take a fixed model as the “frozen” feature extractor, train a lightweight model (probe, usually linear model or MLP) to predict labels.

Frozen: the base model never gets updated when training the lightweight model.

Probing Syntax (Part-of-Speech Tags)



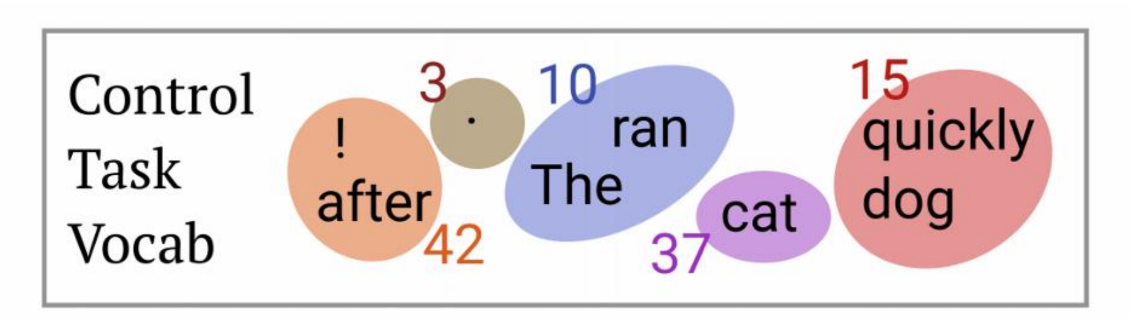
Probing dataset: supervised classification.

Confounding

Q: Does above-chance performance on held out data mean the model encodes part-of-speech knowledge?

A: Not necessarily – the model might just encodes word identity, and the probe learns to group them together.

Solution (control tasks; Hewitt and Liang, 2019): draw conclusion iff. performance on main task is significantly better than that on control task.



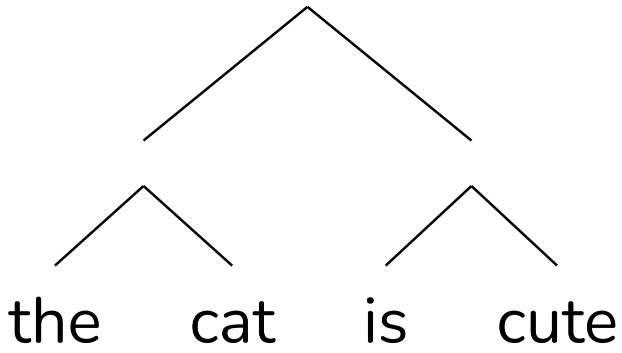
Sentence 1	The	cat	ran	quickly	.
Part-of-speech	DT	NN	VBD	RB	.
Control task	**10**	**37**	**10**	**15**	**3**
Sentence 2	The	dog	ran	after	!
Part-of-speech	DT	NN	VBD	IN	.
Control task	**10**	**15**	**10**	**42**	**42**

Syntax: Constituency

Sentence: *the cat is cute*

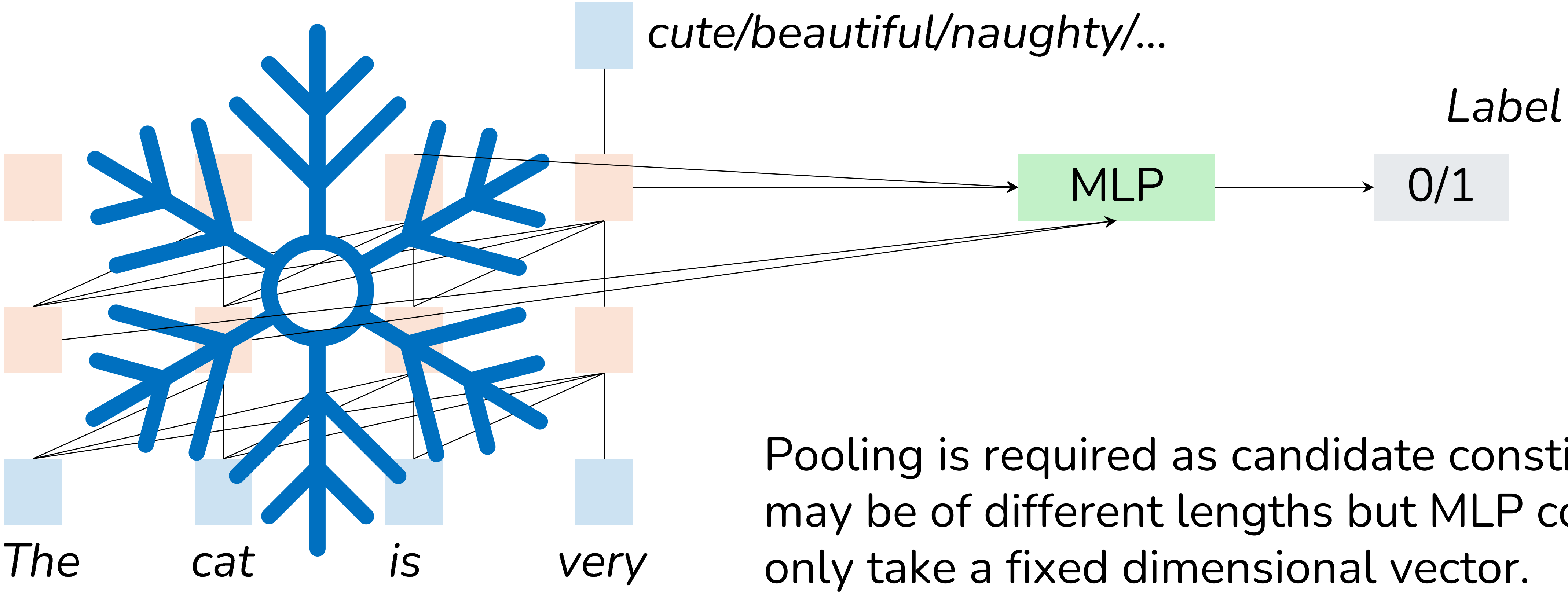
Bracketing: *((the cat) (is cute))*

Tree:



Task: given any span of words -- is it a constituent?

Probing Syntax (Constituency)



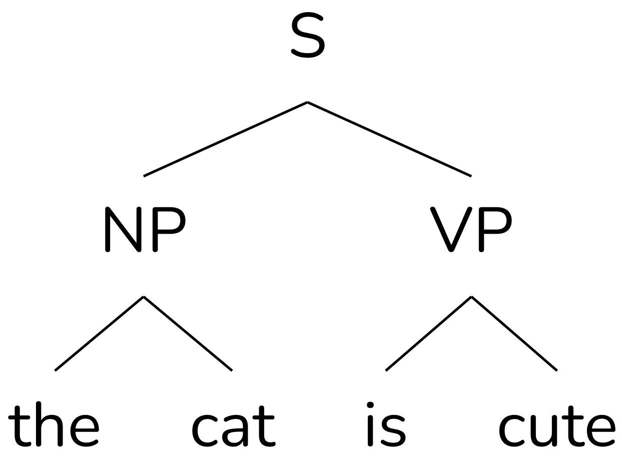
Pooling is required as candidate constituents may be of different lengths but MLP could only take a fixed dimensional vector.

Syntax: Constituent Labels

Sentence: *the cat is cute*

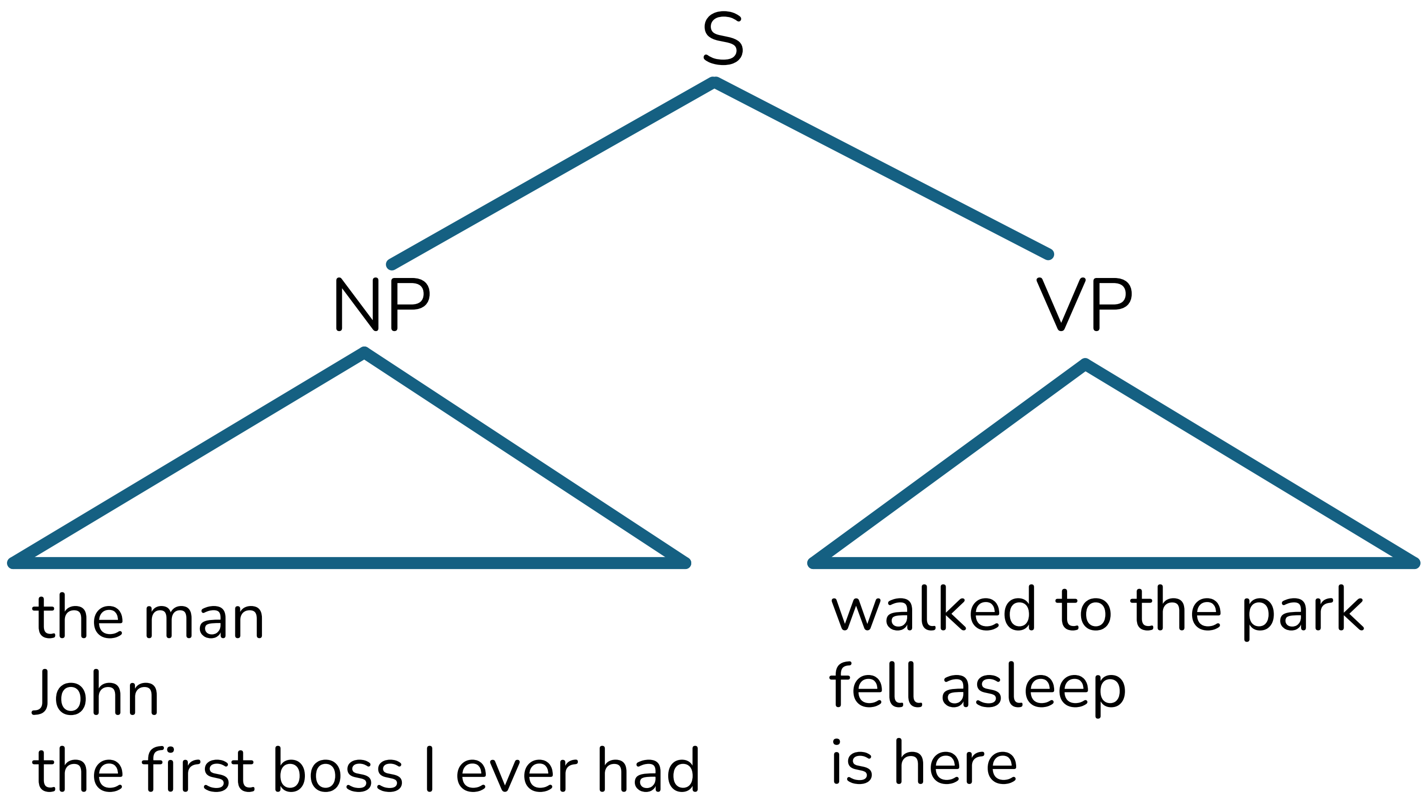
Bracketing: *((the cat) (is cute))*

Tree:

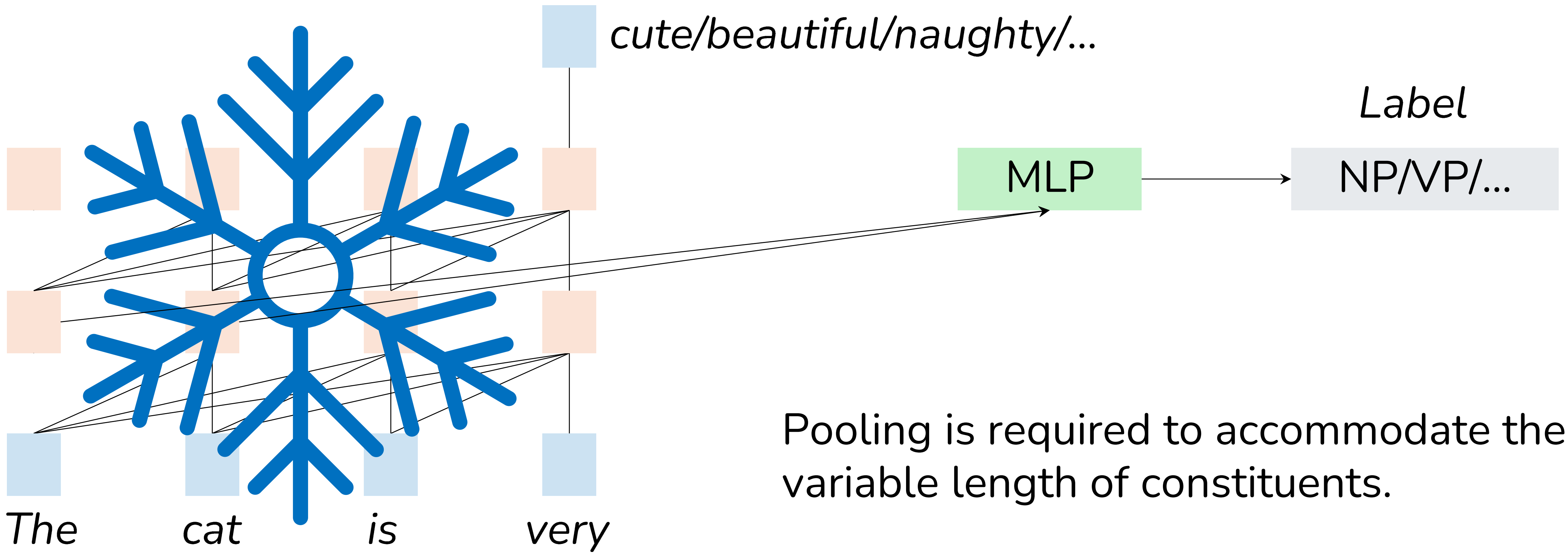


Task: given a constituent, what's the label?

Constituent Labels: Syntactic Substitutability



Probing Syntax (Constituency Labels)



Pooling is required to accommodate the variable length of constituents.

More Tasks

- Does an LM encode numeracy?
- Does an LM “know” if two words refer to the same identity (coreference resolution)?
- Does an LM encode sentence-level information, e.g., topic?
- Does an LM encode knowledge about common sense?

…

BERTology: studying the internals of the BERT model.

Rethinking Probing: Any Issues?

Take a fixed model as the "frozen" feature extractor, train a lightweight model (probe, usually linear model or MLP) to predict labels.

A poor performance may come from:

- The information is not encoded in the model under investigation.
- The information is encoded but not effectively used by the probe.
- The dataset for probe training does not effectively represent the information.

(Belinkov, 2022)

Next

- Basic Syntax: What forms a constituent?
- Context-Free Grammars and Probabilistic Context-Free Grammars