Core Language Models

LLMs: Implications for Linguistics, Cognitive Science & Society

Polina Tsvilodub & Michael Franke, Session 2

Core LLM

- trained on **language modeling objective**
  - predict the next word

"Here is a fragment of text …
According to your **knowledge of the statistics of human language**, what words are likely to come next?

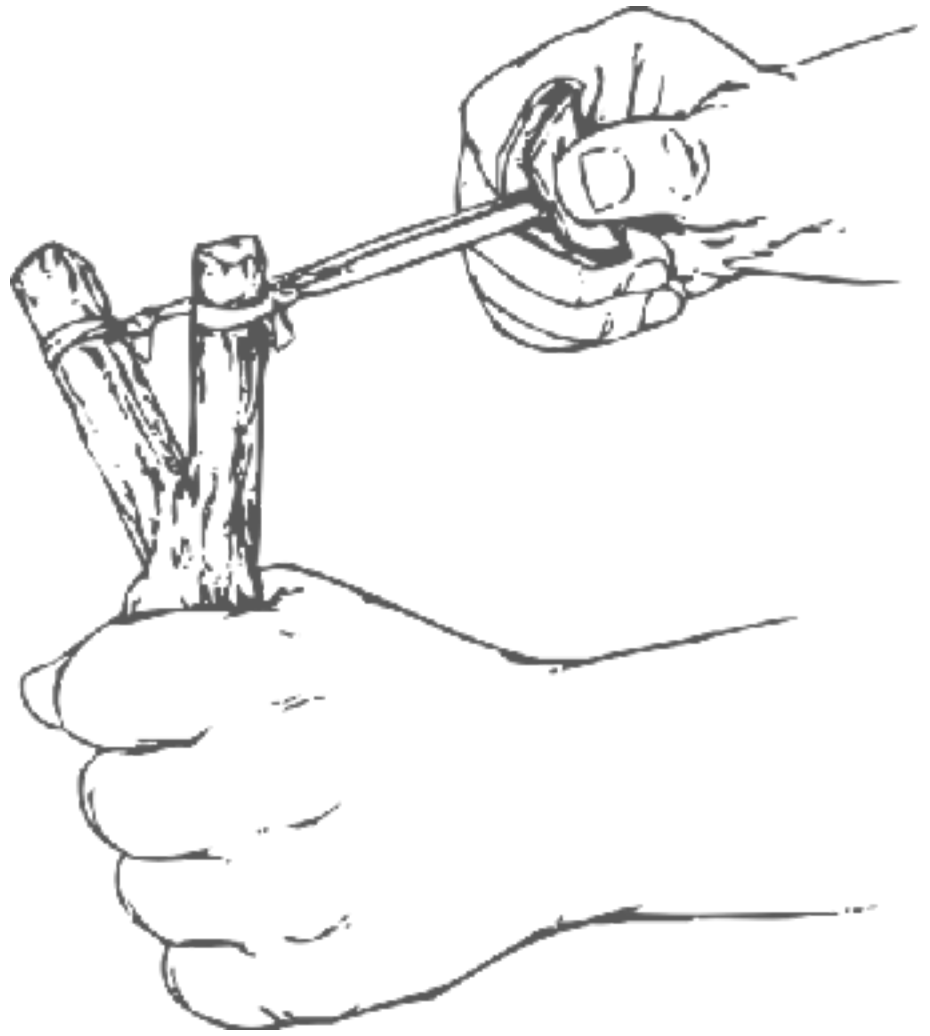
Shanahan (2022)

Prepped LLM

- trained on **usefulness objective**
  - produce text that satisfies user goals

"Here is a fragment of text …
According to your **reward-based conditioning**, what words are likely to trigger positive feedback?"

Learning goals



1. get comfortable with concepts & terminology of **language models**
2. understand **basic architecture of neural LMs**
   a. training (language modeling objective)
   b. decoding
3. become familiar with **transformer models**
   a. self-attention & transformer blocks
   b. heads and layers
   c. uni- vs bidirectional architectures
4. be able to interpret **standard evaluation metrics**

Language Models

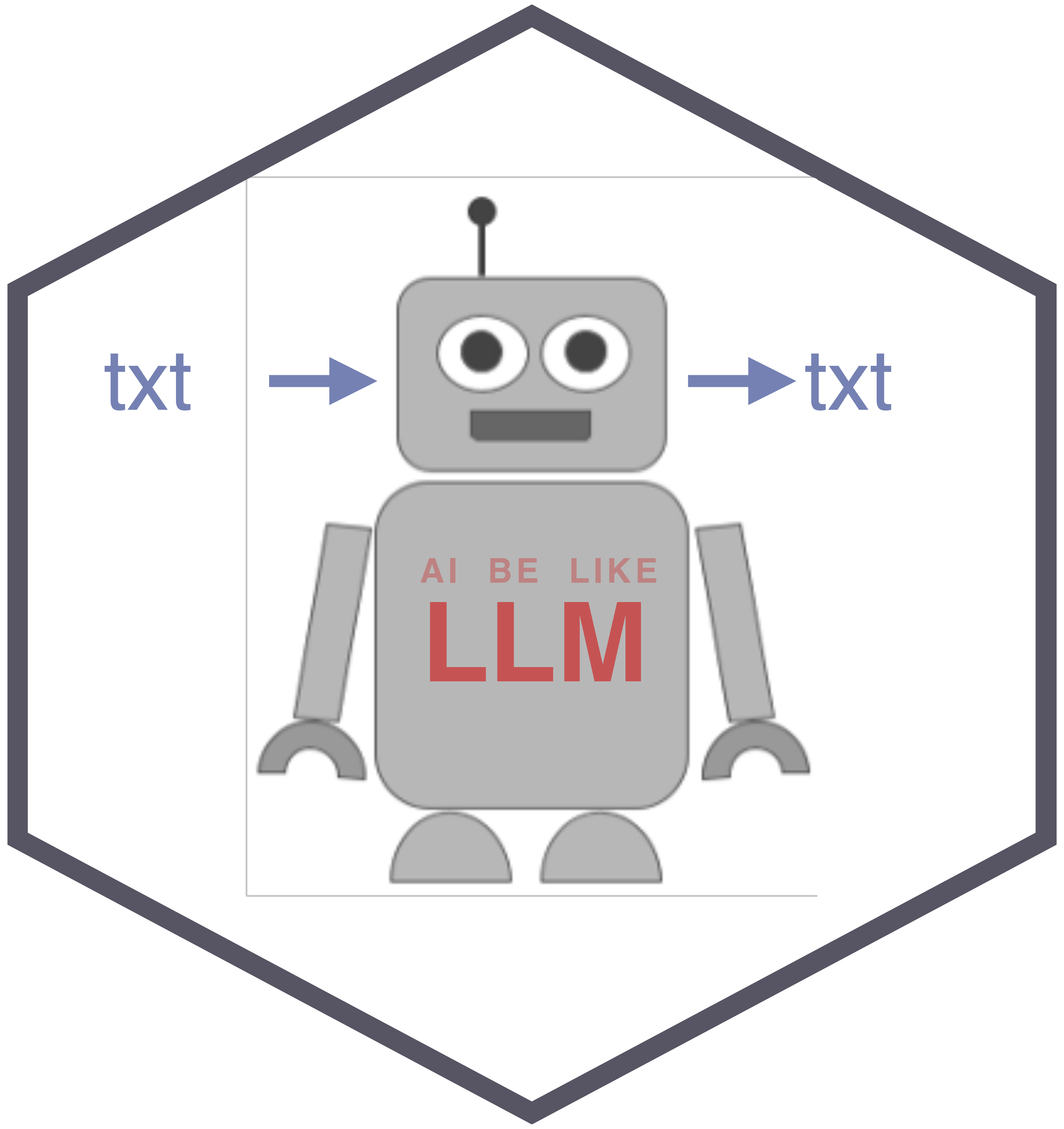
Language model

high-level definition

- let $\mathcal{V}$ be a (finite) **vocabulary**, a set of words
  - we say "words" but these can be characters, sub-words, units …
- let $w_{1:n} = \langle w_1, \dots, w_n \rangle$ be a finite sequence of words
- let S be a the set of all (finite) sequences of words
- let X be a set of input conditions
  - e.g., images, text in a different language …
- a **language model** LM is function that assigns to each input X a probability distribution over S:

  $$LM \; : \; X \mapsto \Delta(S)$$

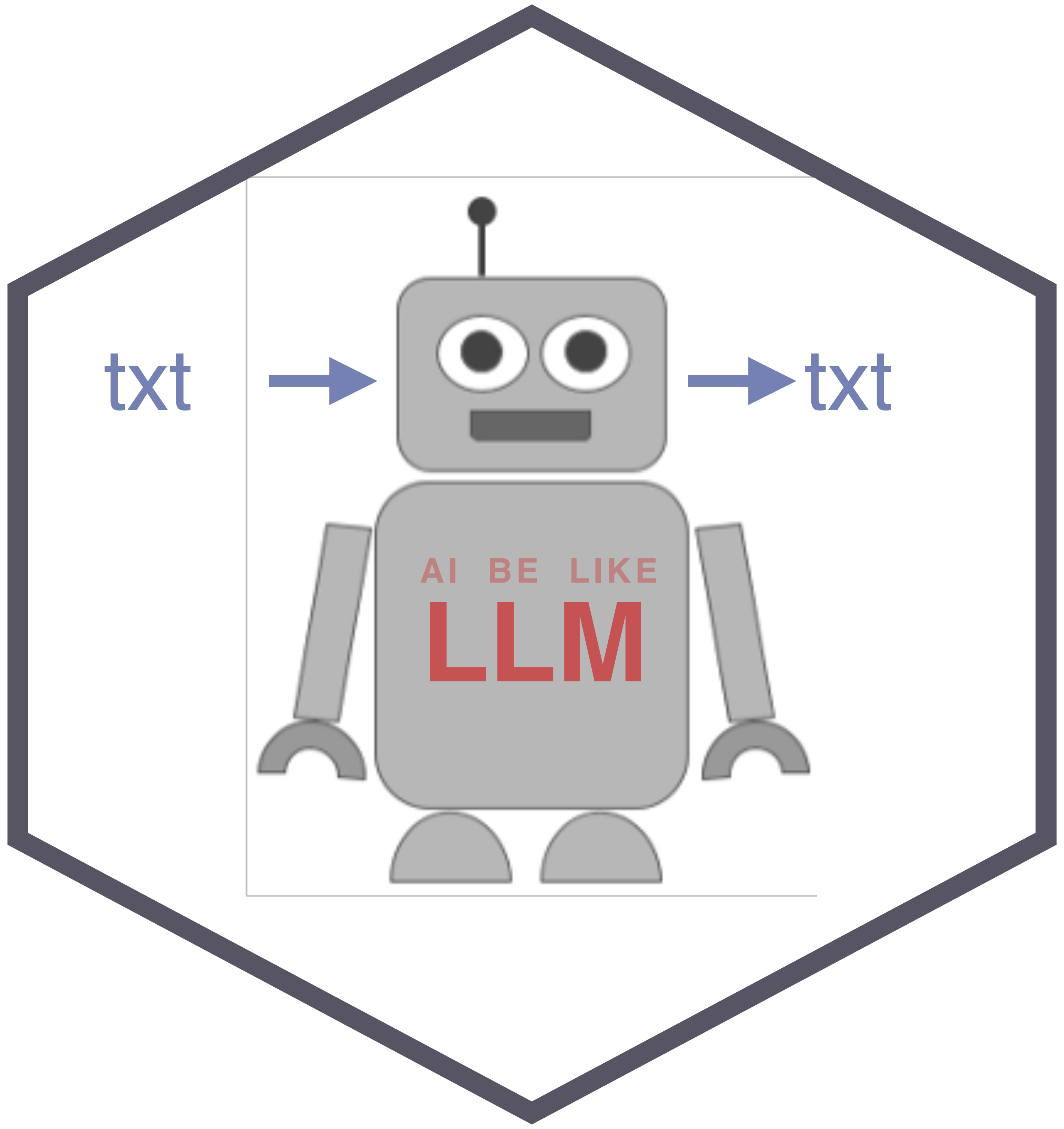
  - if there is only one input in set X, the LM is just a probability distribution over all sequences of words
  - an LM is meant to capture the true relative frequency of occurrence
  - a **neural language model** is an LM realized as a neural network
  - in the following we skip the dependence on X



Language model

left-to-right / causal model

- a **causal language model** is defined as a function that maps an initial sequence of words to a probability distribution over words: $LM \ : \ w_{1:n} \mapsto \Delta(\mathcal{V})$
  - we write $P_{LM}(w_{n+1} \mid w_{1:n})$ for the **next-word probability**
  - the **surprisal** of w_{n+1} after sequence $w_{1:n}$ is $-\log\left(P_{LM}(w_{n+1} \mid w_{1:n})\right)$
- the **sequence probability** follows from the chain rule:
  $P_{LM}(w_{1:n}) = \prod_{i=1}^{n} P_{LM}(w_i \mid w_{1:i-1})$
- measures of **goodness of fit** for observed sequence $w_{1:n}$:
  - **perplexity**:
    $\text{PP}_{LM}(w_{1:n}) = P_{LM}(w_{1:n})^{-\frac{1}{n}}$
  - **average surprisal**:
    $\text{Avg-Surprisal}_{LM}(w_{1:n}) = -\frac{1}{n} \log P_{LM}(w_{1:n})$



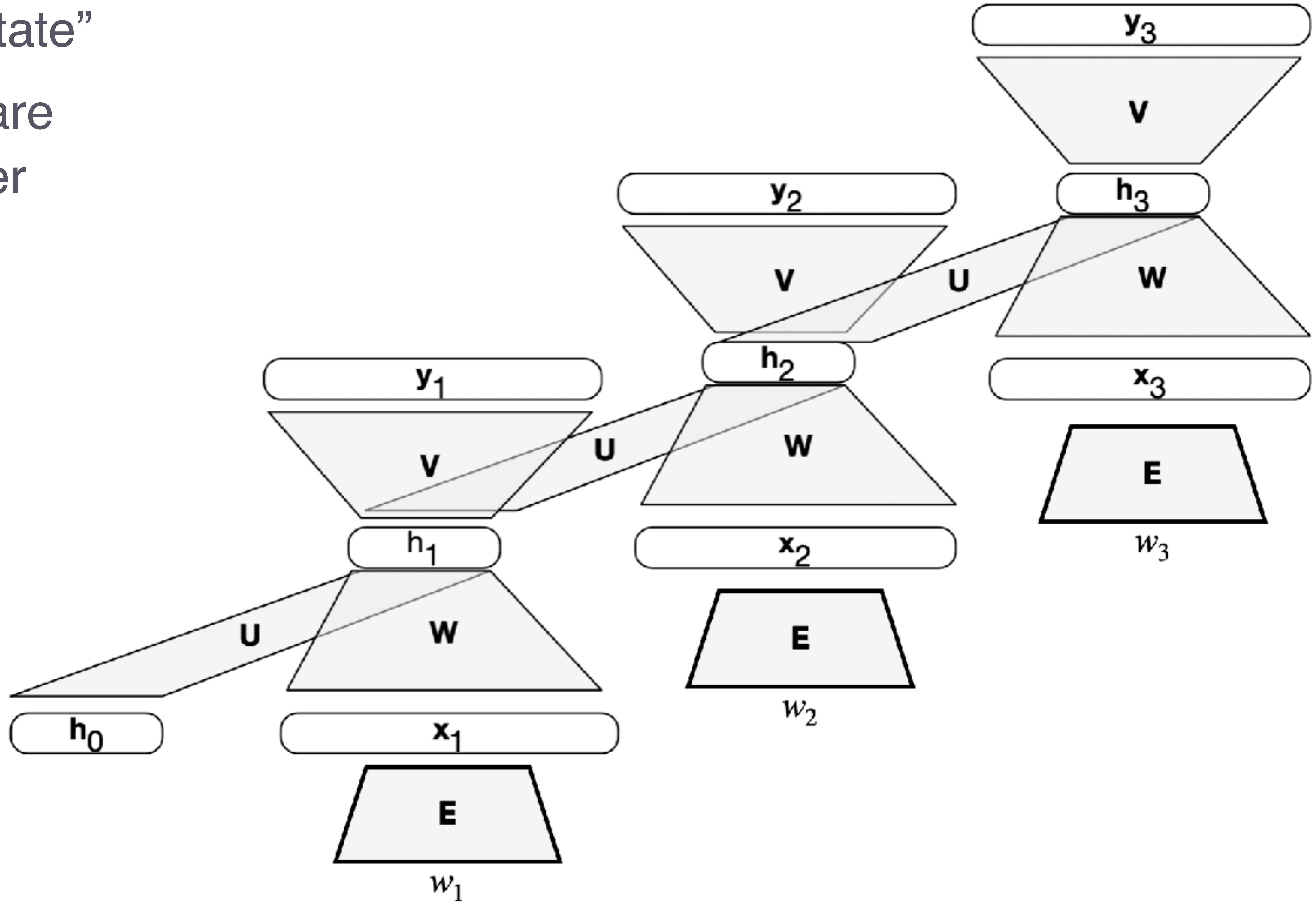
$\log \text{PP}_M(w_{1:n}) = \text{Avg-Surprisal}_M(w_{1:n})$

Recurrent Neural Networks

RNN-based language model

hidden layer as a "memory state"

- hidden layer is a "memory state"
- predictions (at each token) are derived from the hidden layer
- applicable to:
  - next-word prediction
  - part-of-speech prediction
  - sentiment analysis
  - …

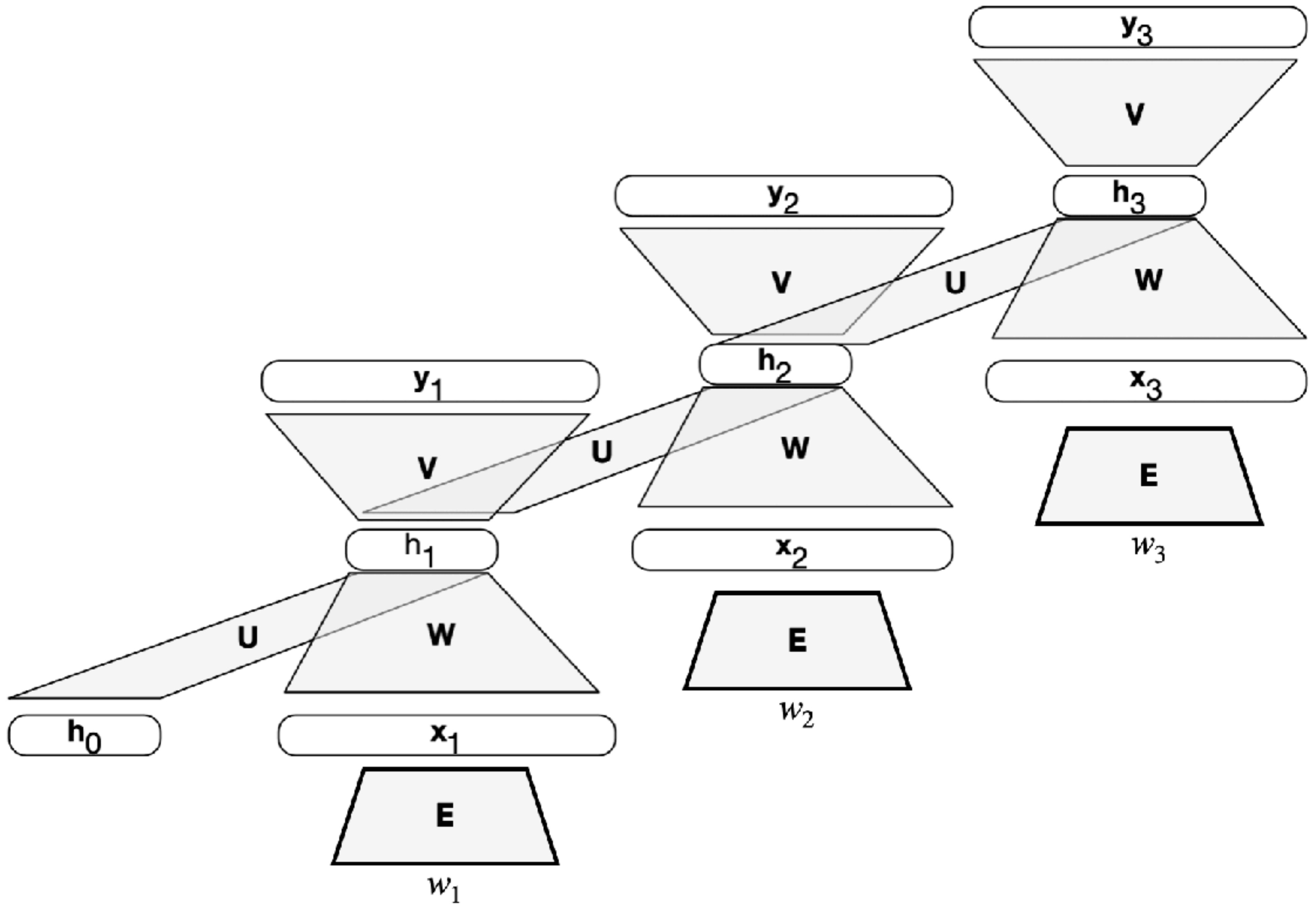


RNN-based language model

one of many similar architectures

- dimensions:
  - n_V : # of words in vocabulary
  - n_h : # units in hidden layer
  - n_x : length of input $\mathbf{x}$ (word embedding)
- what is what?
  - $\mathbf{w}_t \in \mathbb{R}^{n_V}$: one-hot vector representing word $\mathbf{w}_t$
  - $\mathbf{x}_t \in \mathbb{R}^{n_x}$: word embedding of word $\mathbf{w}_t$
  - $\mathbf{h}_t \in \mathbb{R}^{n_h}$: hidden layer activation at time t (with $\mathbf{h}_0 = 0$)
  - $\mathbf{y}_t \in \Delta(\mathscr{V})$: probability distribution over words
  - $f \in \{\sigma, \tanh, \dots\}$: activation function (as usual)
  - $\mathbf{U} \in \mathbb{R}^{n_h \times n_h}$: mapping hidden-to-hidden
  - $\mathbf{V} \in \mathbb{R}^{n_V \times n_h}$: mapping hidden-to-word
  - $\mathbf{E} \in \mathbb{R}^{n_x \times n_V}$: mapping word-to-embedding
  - $\mathbf{W} \in \mathbb{R}^{n_h \times n_x}$: mapping embedding-to-hidden

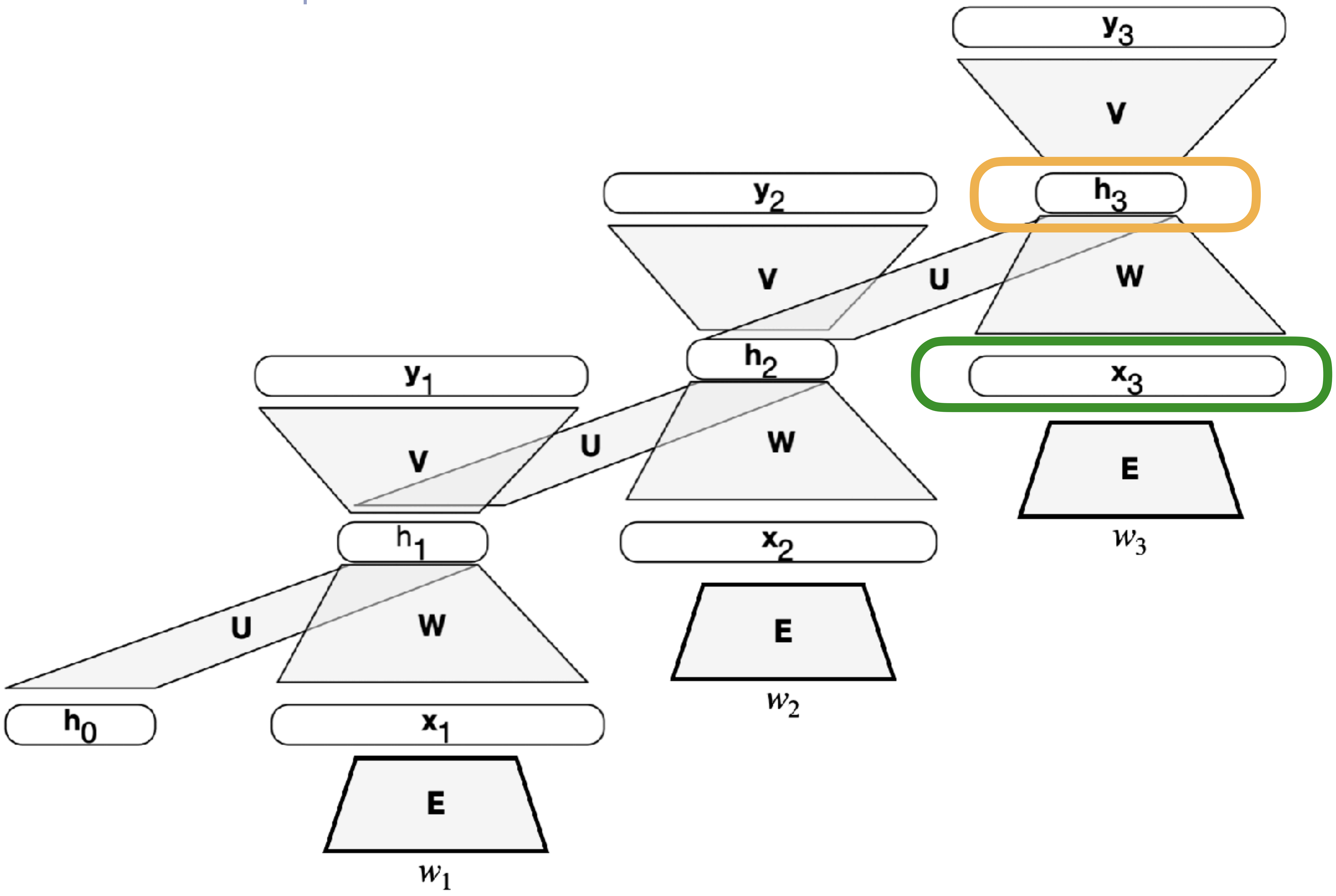
- definition (forward pass):
  - $\mathbf{x}_t = \mathbf{E}\mathbf{w}_t$
  - $\mathbf{h}_t = f\left[\mathbf{U}\mathbf{h}_t + \mathbf{W}\mathbf{x}_t\right]$
  - $\mathbf{y}_t = \text{softmax}(\mathbf{V}\mathbf{h}_t)$



based on Jurafsky & Martin “NLP” book draft

Embeddings

for words and sequences



sequence embedding for w_1, w_2, w_3

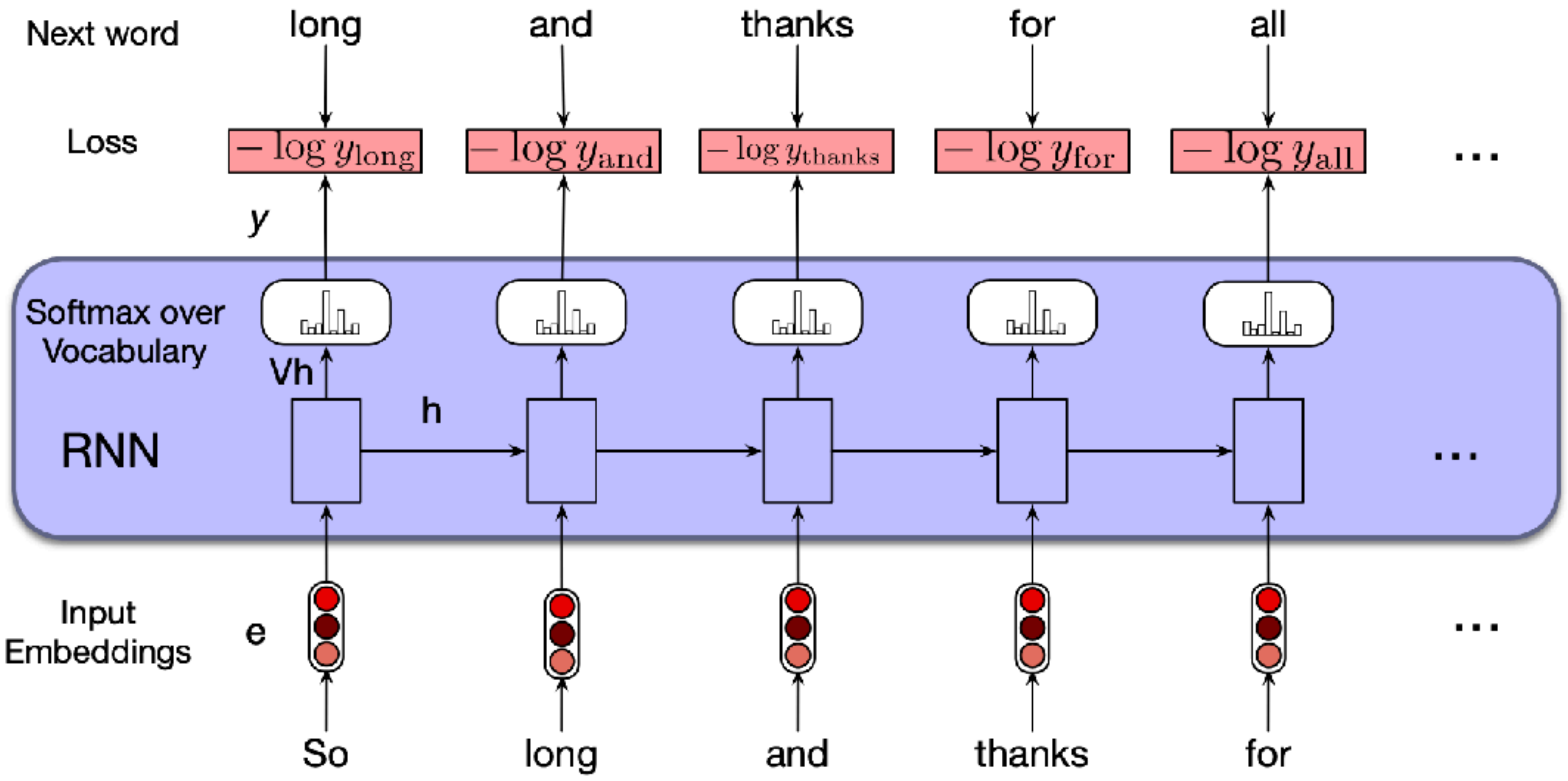
word embedding for w_3

Training

Training RNNs

using teacher forcing & next-word surprisal

- **teacher forcing**
  - predict each next word given the preceding input (not the model-generated sequence)
- **next-word surprisal**
  - loss function is (average) next-word surprisal
  - NB: surprisal = cross-entropy if training item is non-stochastic



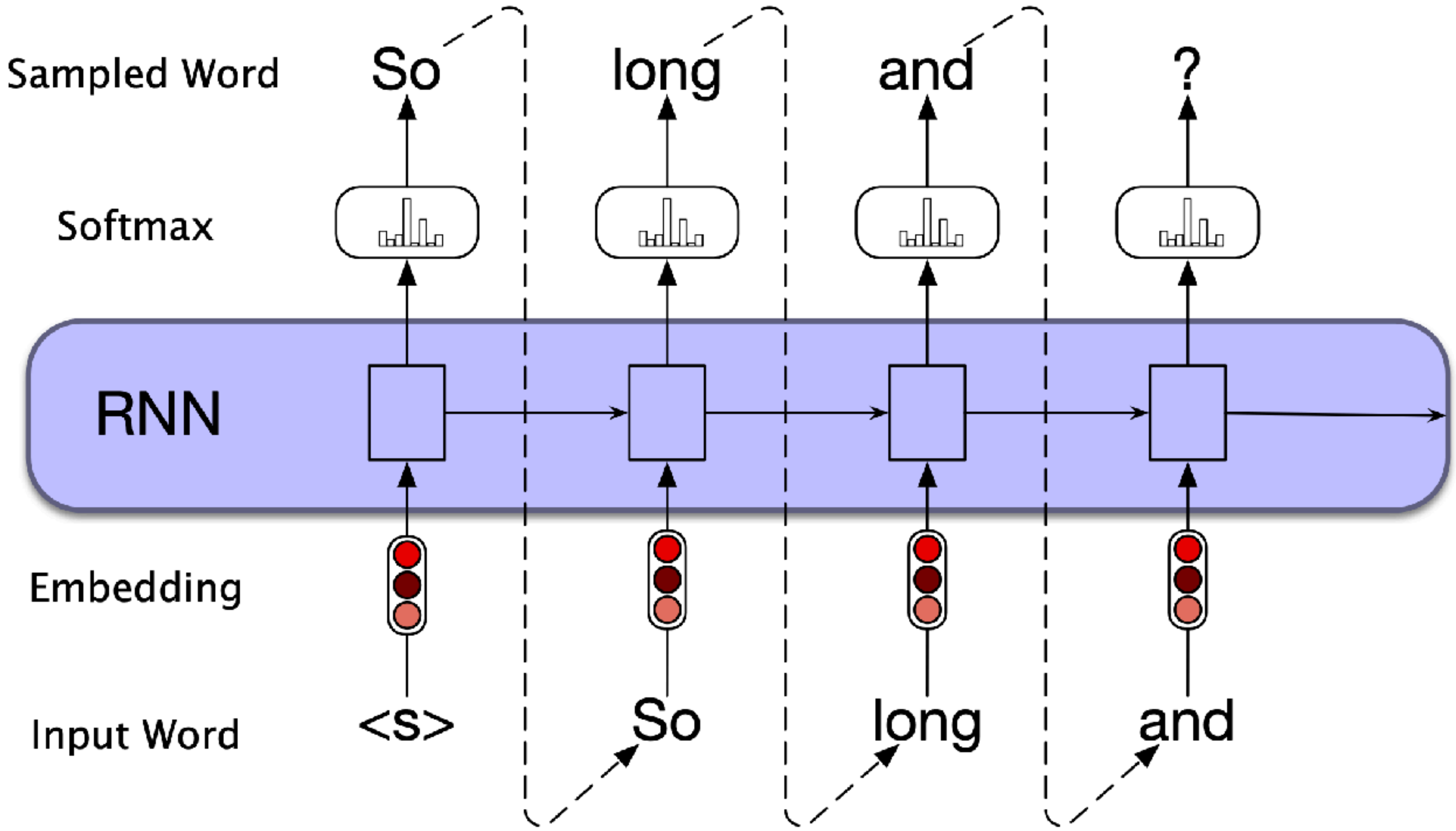
Common training regimes

- **teacher forcing**
  - LM is fed true word sequence
  - training signal is next-word assigned to true word
- **autoregressive training** (aka free-running mode)
  - LM autoregressively generates a sequence
  - training signal is next-word probability assigned to true word
- **curriculum learning** (aka scheduled sampling)
  - combine teacher-forced and autoregressive training
  - start with mostly teacher forcing, then increase amount of autoregressive training
- **professor forcing**
  - combines teacher forcing with adversarial training
  - generative adversarial network GAN is trained to discriminate (autoregressive) predictions from actual data
  - LM is trained to minimized this discriminability
- **decoding-based**
  - use prediction function (decoding scheme) to optimize based on *actual* output

Decoding

Autoregressive generation

left-to-right / causal model



Common decoding schemes

based on next-word probability $P(w_{i+1} \mid w_{1:i})$

- **pure sampling**
  - next word is sampled from next-word probability distribution: $w_{i+1} \sim P(\cdot \mid w_{1:i})$
- **greedy decoding**
  - next word is word with highest probability: $w_{i+1} = \arg\max_{w'} P(w' \mid w_{1:i})$
- **softmax sampling**
  - next word is sampled from softmax of next-word probability distribution: $w_{i+1} \sim SM_{\alpha}\left(P(\cdot \mid w_{1:i})\right)$
- **top-k sampling**
  - next word is sampled from next-word prob. distribution after restricting to the k most likely words
- **top-p sampling (=nucleus sampling)**
  - next word is sampled from next-word prob. distribution after restricting to the smallest set of the most likely words which together comprise at least next-word probability p
- **beam search**
  - greedily construct sequences of best *k* words

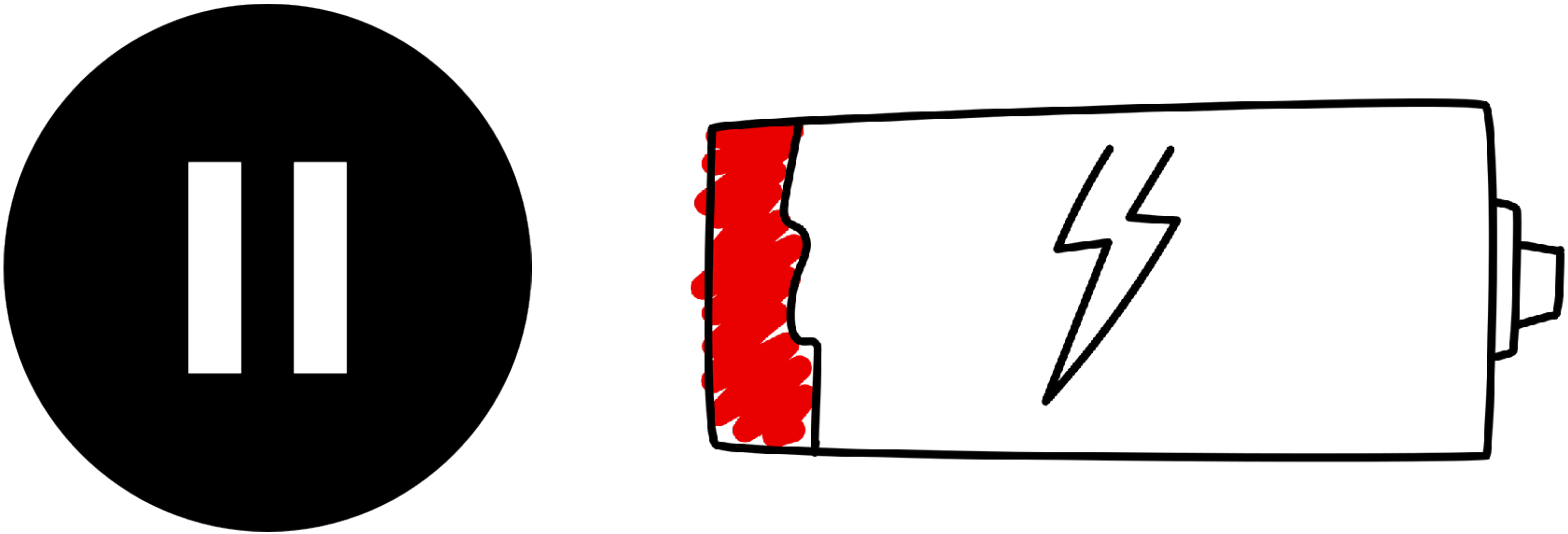
demo

colab notebook for different decoding schemes

Summary

language models

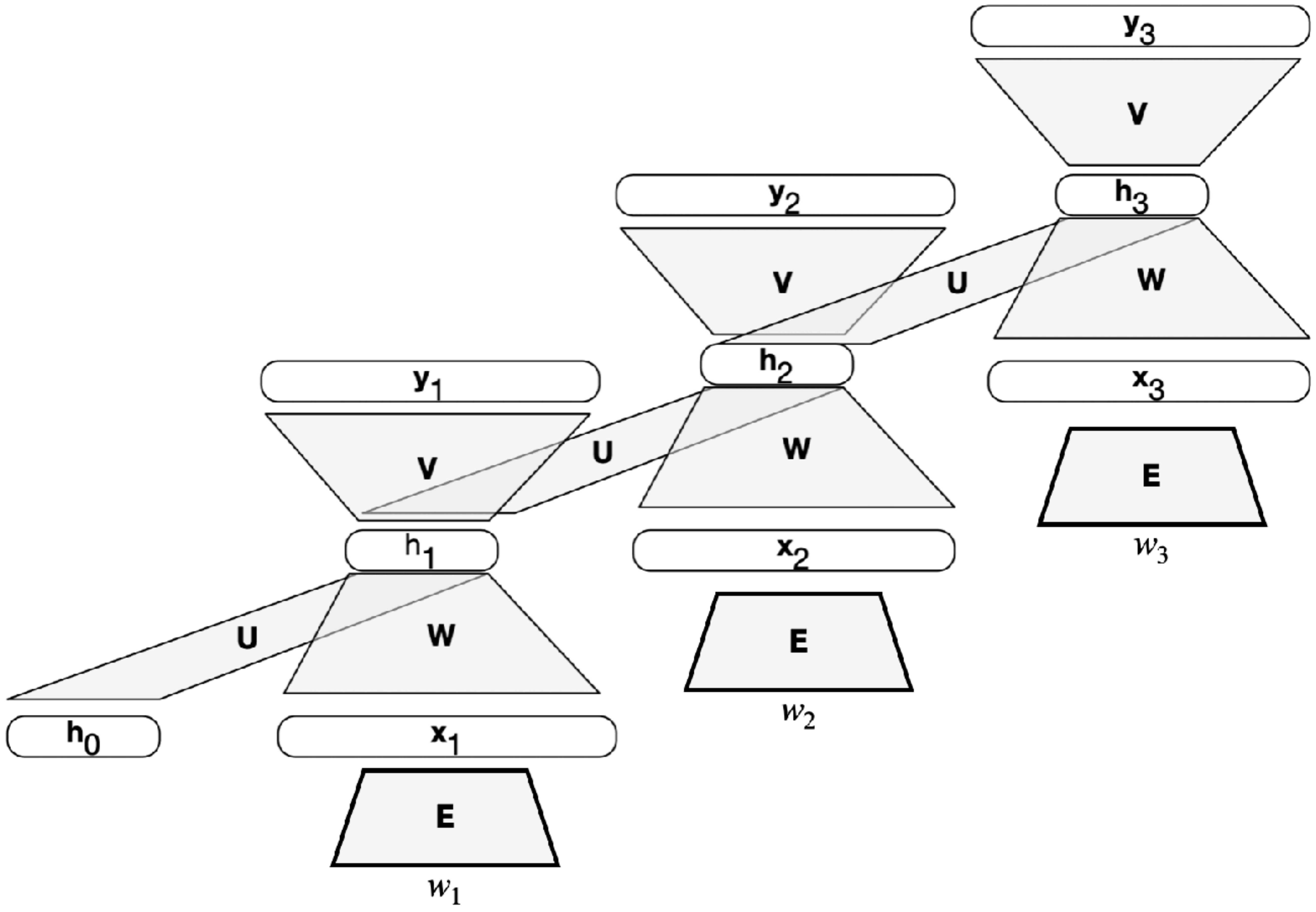
- language models approximate true $\Delta(S)$
- causal LMs define next-word probabilities
- training
  - language modeling objective: maximize next-word probability
  - teacher forcing: supply everything except the next word to be predicted
- decoding
  - different stochastic sampling regimes
  - beam search is best but costly



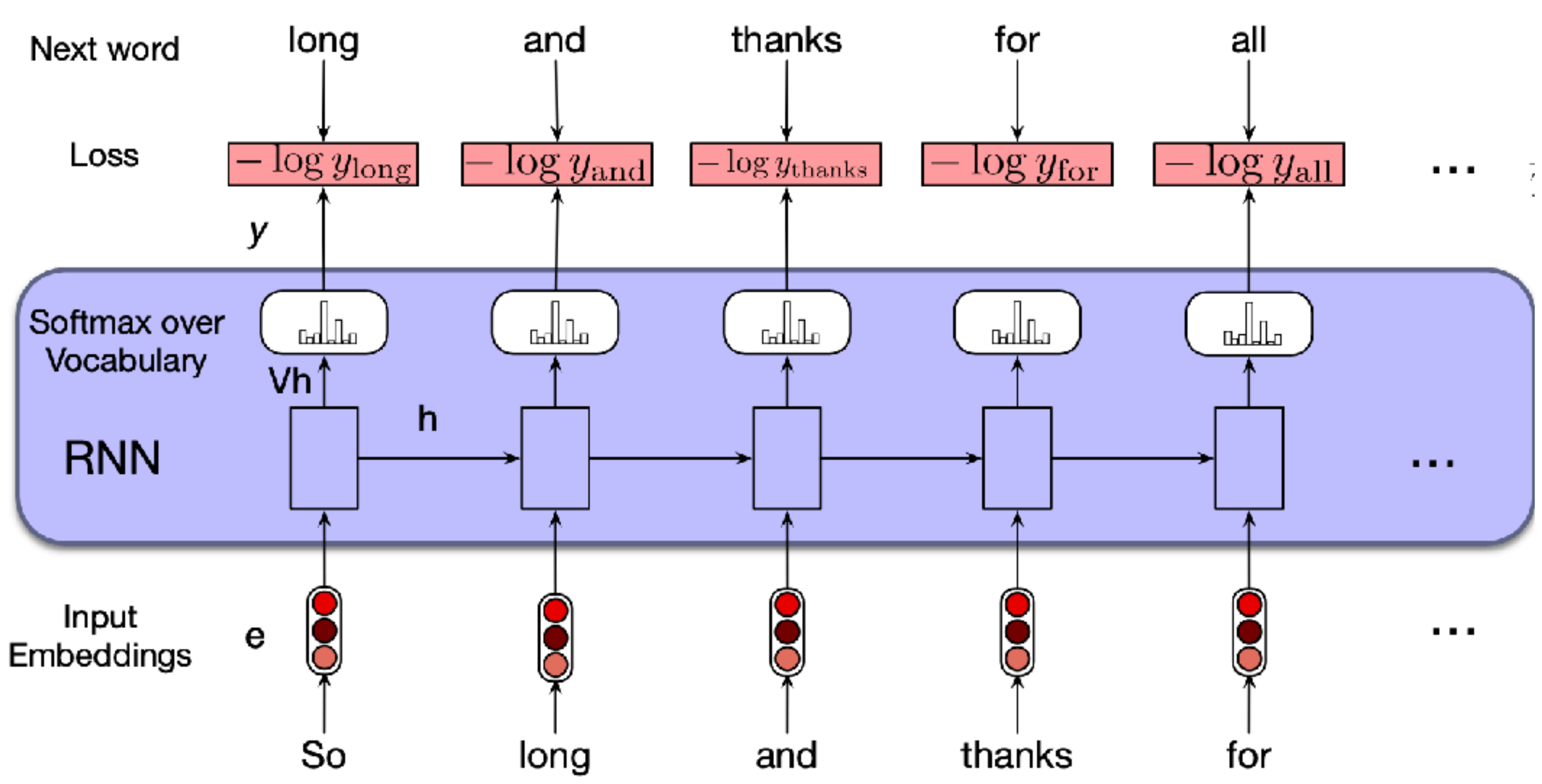
Transformers
& self-attention

Problems with RNNs

- **conceptual problem**
  - two-fold role of hidden state:
    - memory for past sequence
    - recommend what to do now
- **technical problem**
  - vanishing gradients for long past input
    - partial remedy: bidirectional RNNs

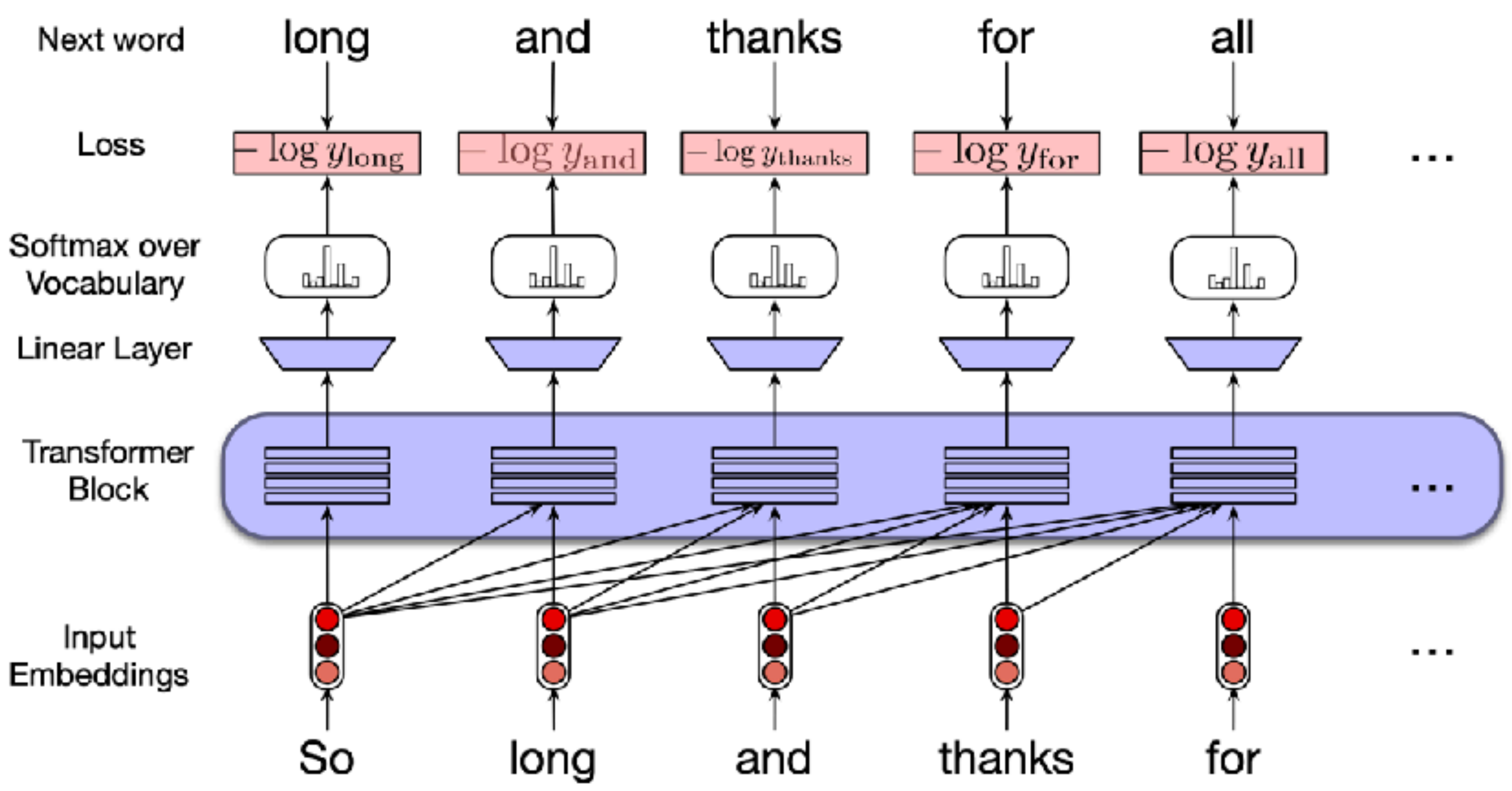


RNN



Transformer

left-to-right architecture



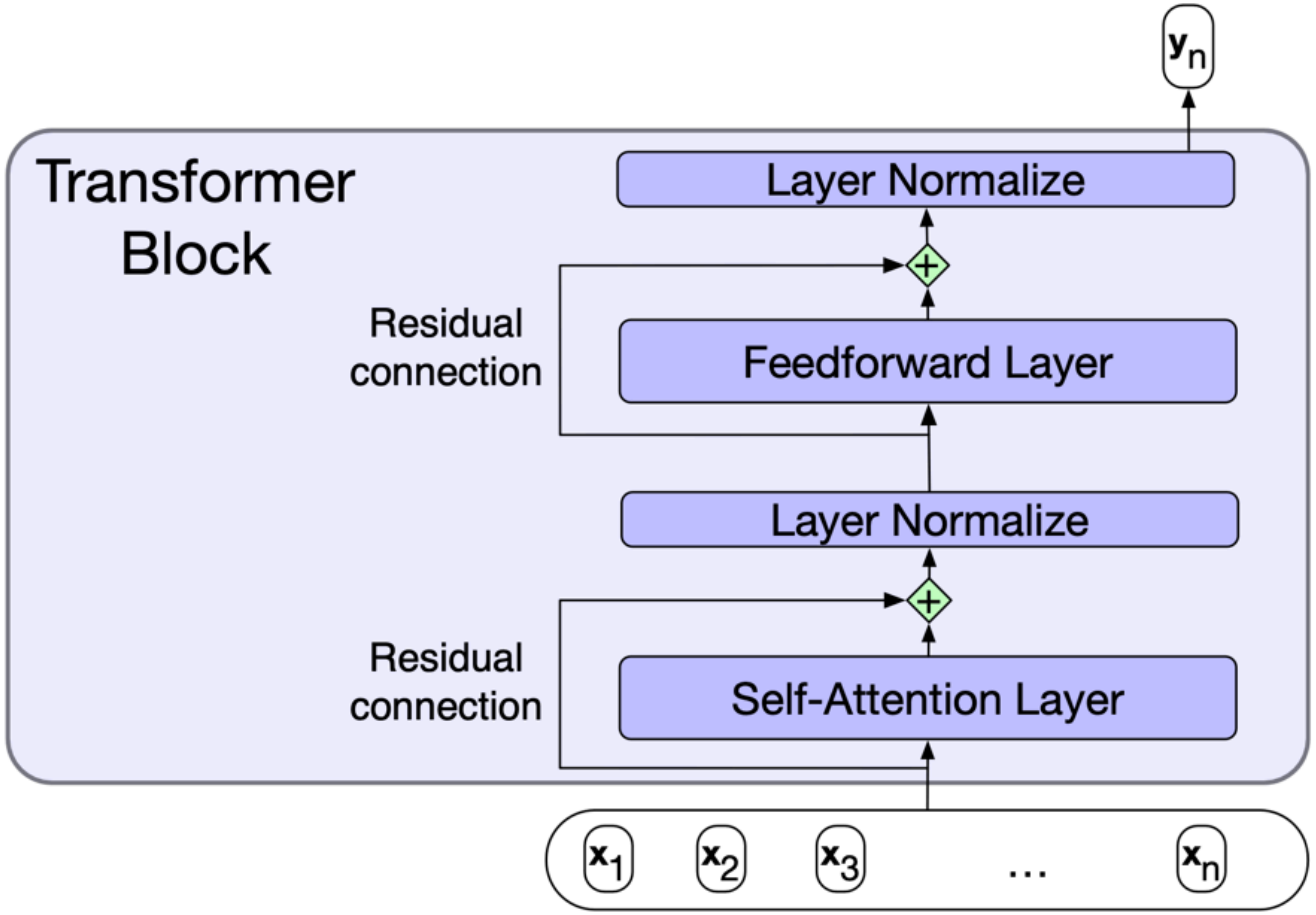
Transformer blocks

- layer normalization:

$$\text{LayerNorm}(\mathbf{x}) = \gamma \ \text{z-score}(\mathbf{x}) + \beta$$

$$\text{z-score}(\mathbf{x}) = \frac{\mathbf{x} - \text{mean}(\mathbf{x})}{\text{SD}(\mathbf{x})}$$

- residual connection
  - facilitates learning
- self-attention layer
  - key novel innovation



Self-attention layer

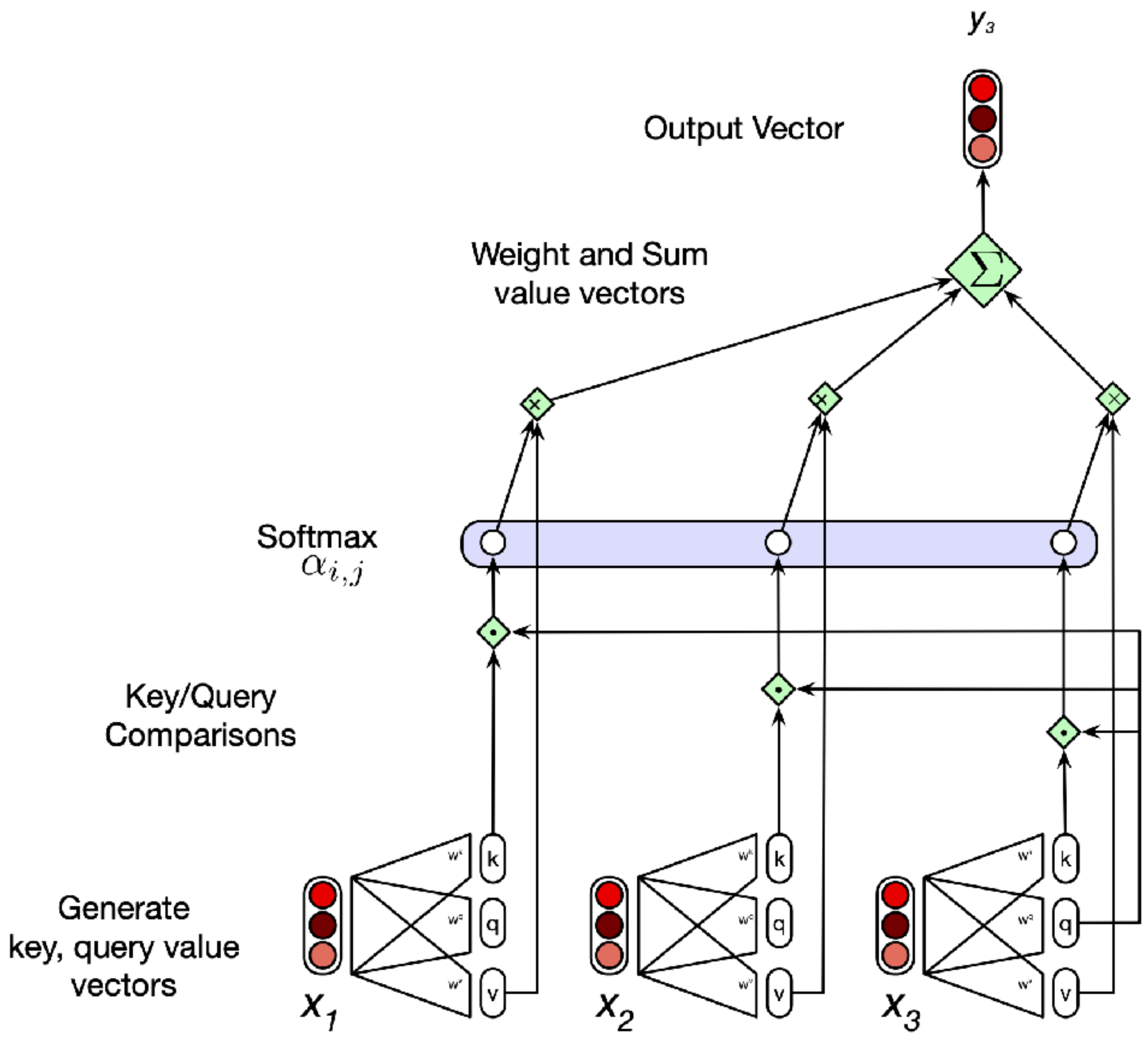
- **output**

$$\mathbf{y}_i = \sum_{j \le i} \alpha_{ij} \mathbf{v}_j$$

- **weight score**

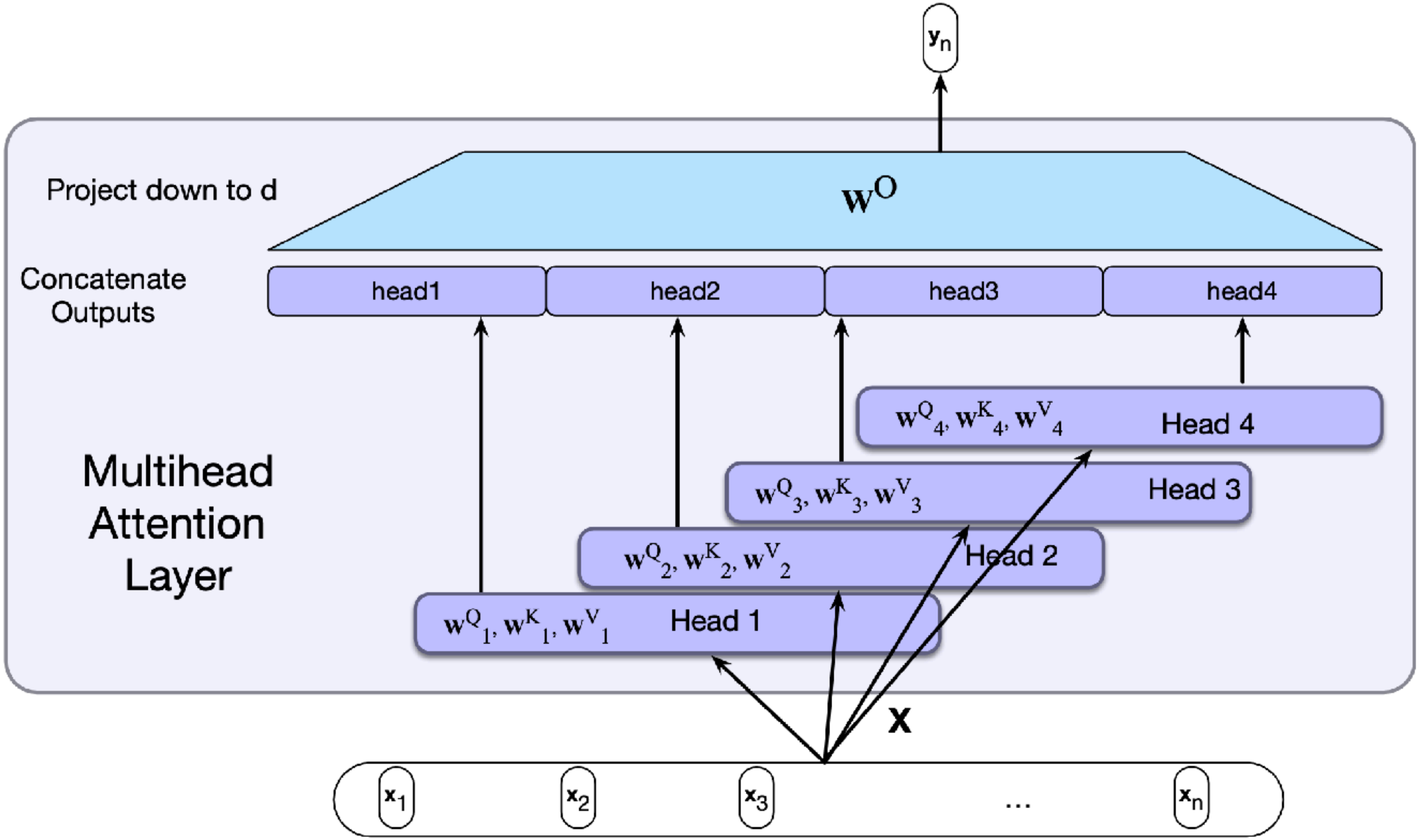
$$\alpha_{i,j} = \frac{\exp(\mathbf{q}_i \cdot \mathbf{k}_j)}{\sum_{j' \le i} \exp(\mathbf{q}_i \cdot \mathbf{k}_{j'})}$$

- three vectors for each input vector x_i
  1. **query:** which info to extract from context
     $\mathbf{q}_i = \mathbf{W}^Q \mathbf{x}_i$
  2. **key:** which info to provide for later
     $\mathbf{k}_i = \mathbf{W}^K \mathbf{x}_i$
  3. **value:** what output to choose
     $\mathbf{v}_i = \mathbf{W}^V \mathbf{x}_i$

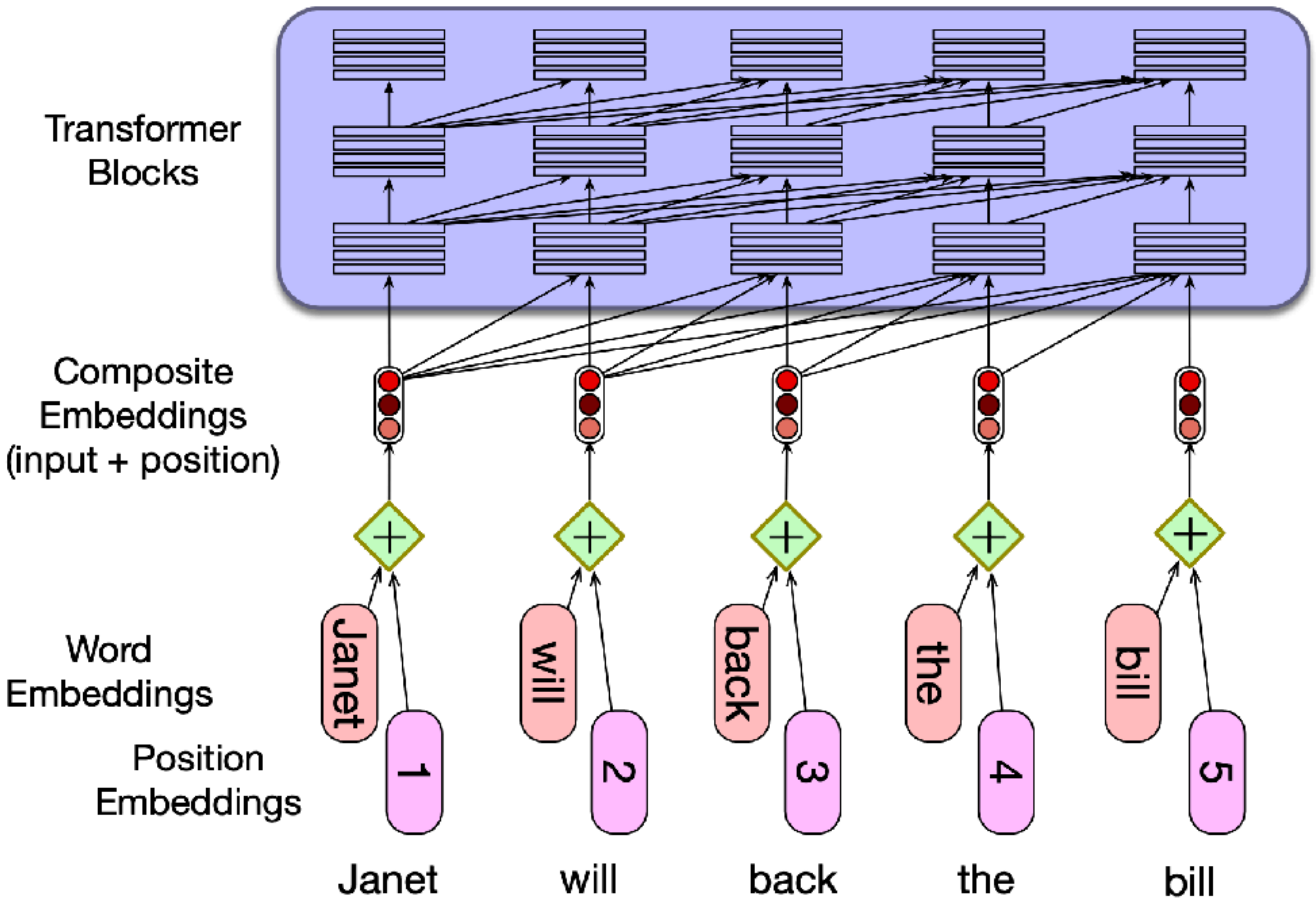


Vaswani et al. (2017)

Multihead attention layer



Positional encoding



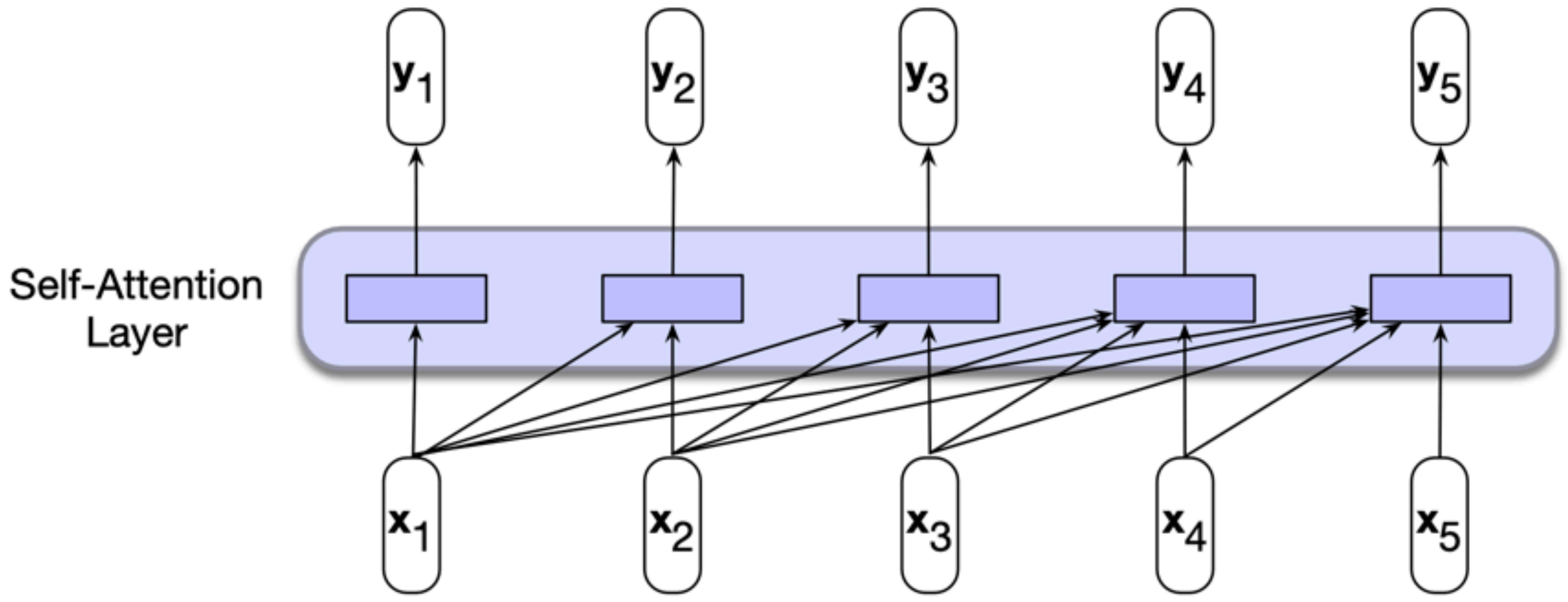
$$\vec{p_t} = \begin{bmatrix} \sin(\omega_1 . t) \\ \cos(\omega_1 . t) \\ \\ \sin(\omega_2 . t) \\ \cos(\omega_2 . t) \\ \\ \vdots \\ \\ \sin(\omega_{d/2} . t) \\ \cos(\omega_{d/2} . t) \end{bmatrix}_{d\times 1} \qquad \omega_k = \frac{1}{10000^{2k/d}}$$

Summary
transformers

- access information from the full (left) input
- use self-attention to offer and retrieve relevant information
- stack transformer blocks to enable functional feature separation

Language Model Architectures

Causal LM

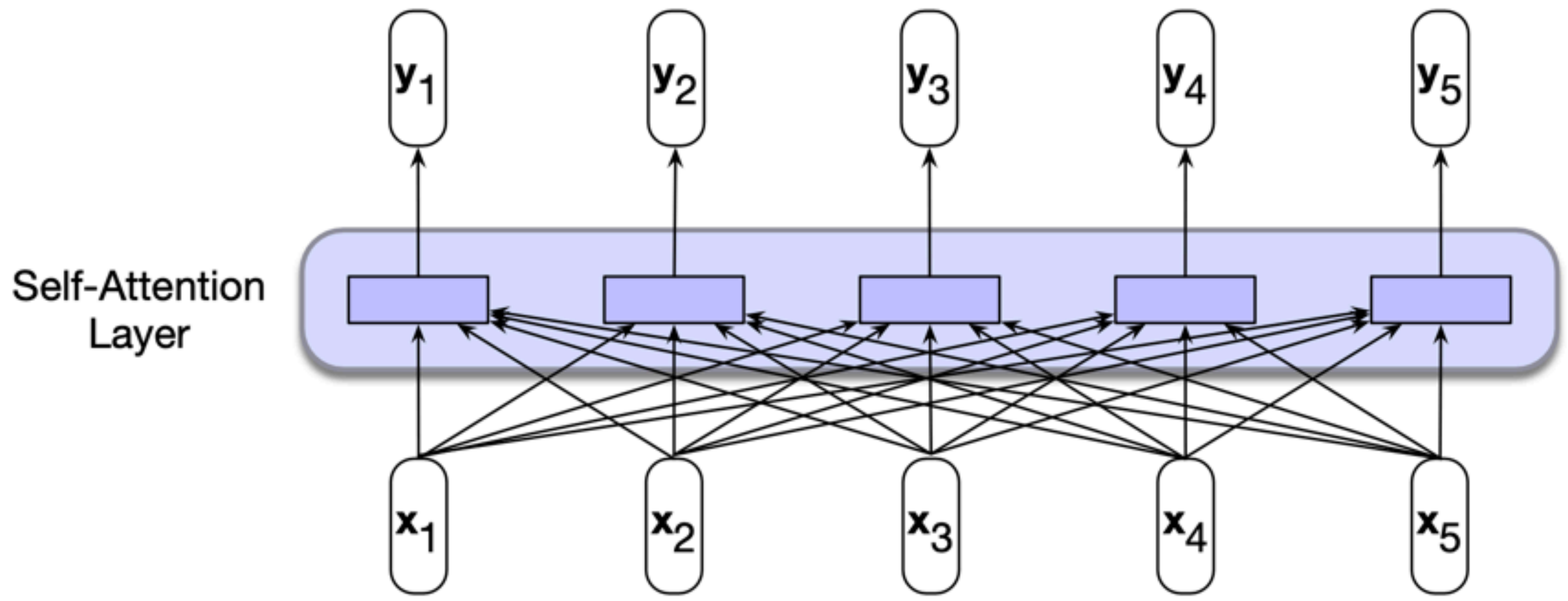


computation for input $\mathbf{x}_1, \ldots, \mathbf{x}_3$ blind to $\mathbf{x}_4$ and $\mathbf{x}_5$

$\mathbf{y}_5$ is embedding for input $\mathbf{x}_1, \ldots, \mathbf{x}_5$

$\mathbf{y}_5$ is a "left-contextual embedding"

Bidirectional encoder



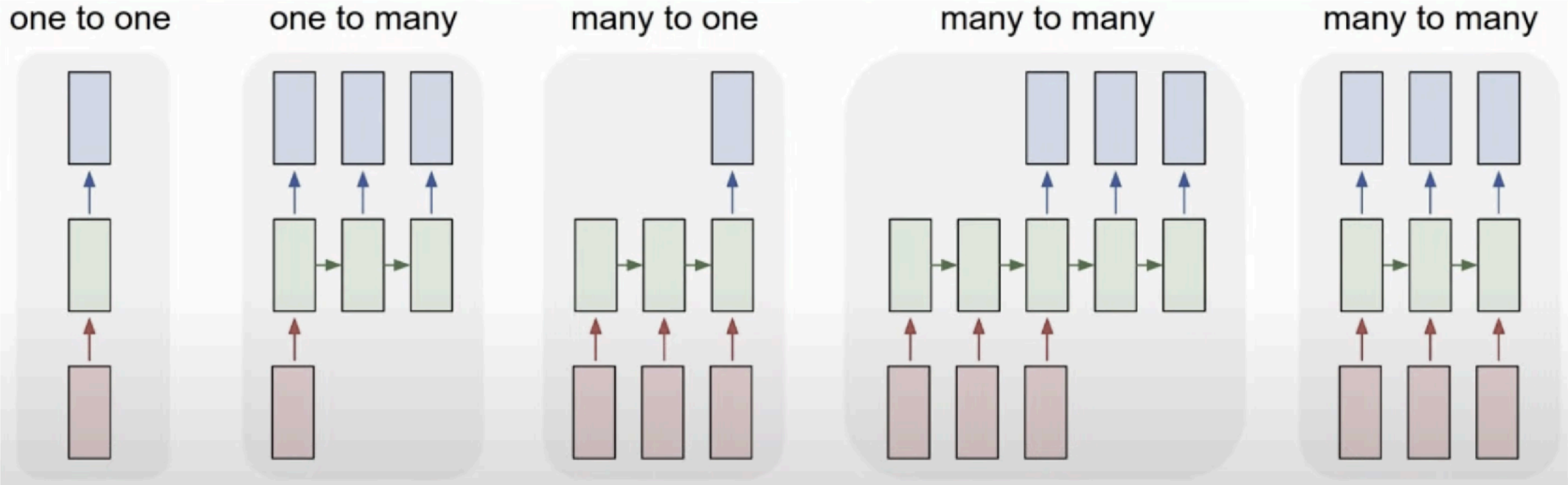
computation for input $\mathbf{x}_1, \ldots, \mathbf{x}_3$ sees $\mathbf{x}_4$ and $\mathbf{x}_5$

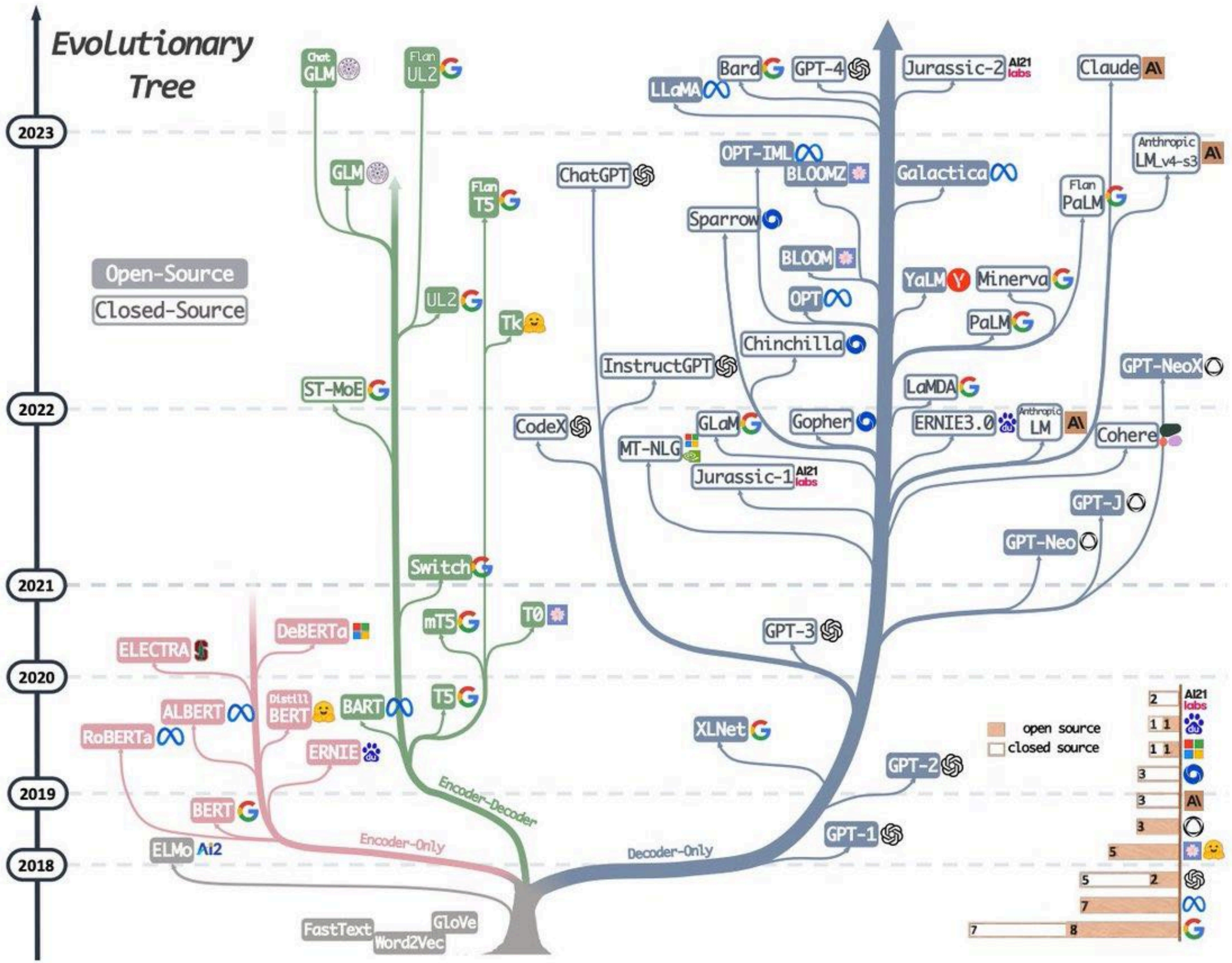
$\mathbf{y}_1, \ldots, \mathbf{y}_5$ is embedding for input $\mathbf{x}_1, \ldots, \mathbf{x}_5$

$\mathbf{y}_i$ are bidirectional "contextual embeddings"

Different kinds of sequence processing models

sequence as input and/or (simultaneous) output





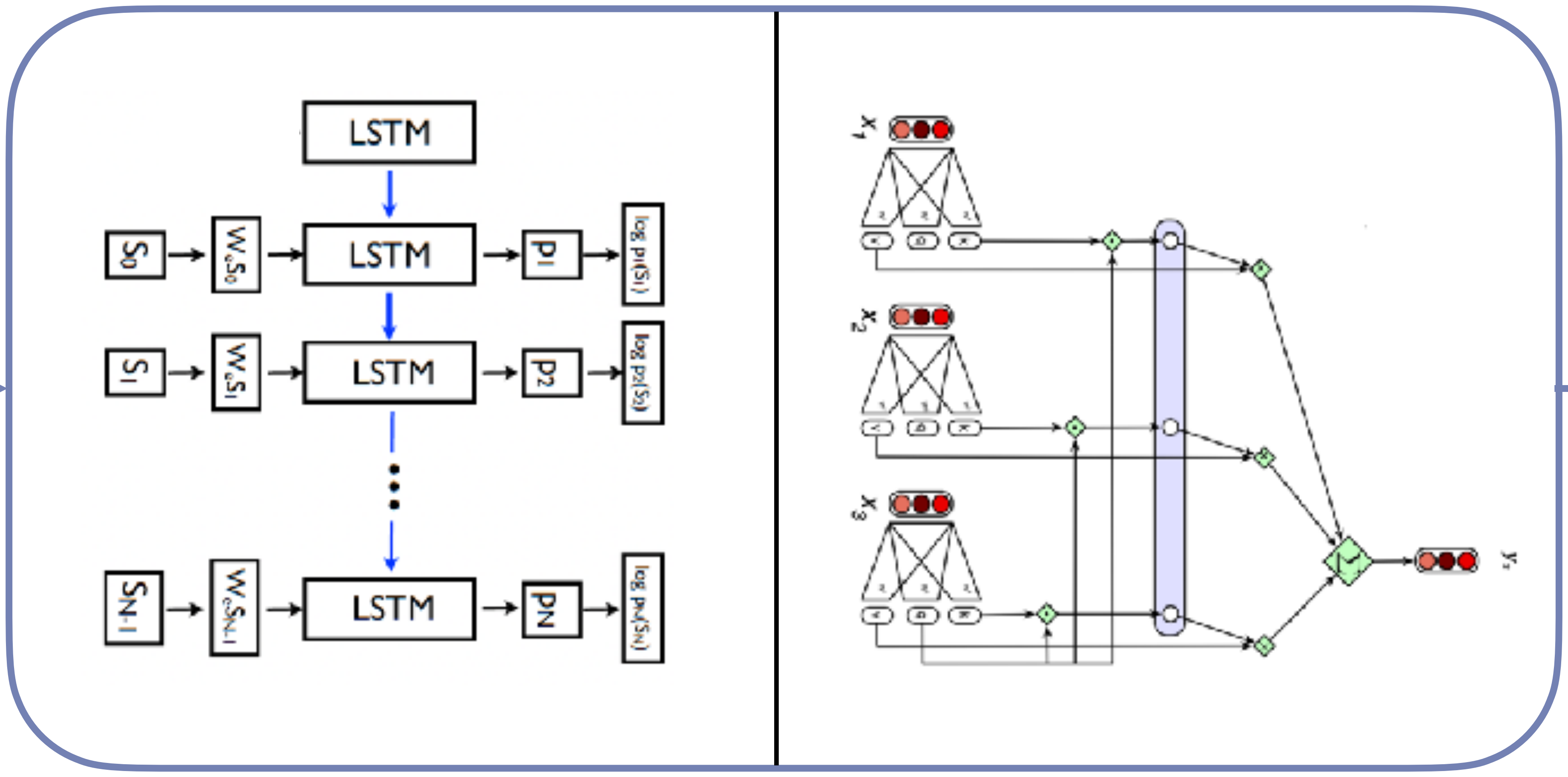
Evolutionary Tree
2023
2022
2021
2020
2019
2018
Open-Source
Closed-Source
Chat GLM
Flan UL2
GLM
Flan T5
UL2
Tk
ST-MoE
Switch
mT5
T0
T5
BART
ELECTRA
DeBERTa
ALBERT
Distill BERT
RoBERTa
ERNIE
BERT
ELMo
Ai2
Encoder-Only
Encoder-Decoder
Decoder-Only
FastText
Word2Vec
GloVe
LLaMA
Bard
GPT-4
Jurassic-2
AI21 labs
Claude
ChatGPT
OPT-IML
BLOOMZ
Galactica
Anthropic LM_v4-s3
Flan PaLM
Sparrow
BLOOM
YaLM
Minerva
OPT
PaLM
Chinchilla
InstructGPT
LaMDA
GPT-NeoX
CodeX
GLaM
Gopher
ERNIE3.0
Anthropic LM
Cohere
MT-NLG
Jurassic-1
GPT-J
GPT-Neo
GPT-3
XLNet
GPT-2
GPT-1
open source
closed source
2
1 1
1 1
3
3
3
5
5 2
7
7 8

source here

Evaluating LMs
Benchmarks & Metrics

Evaluating LMs

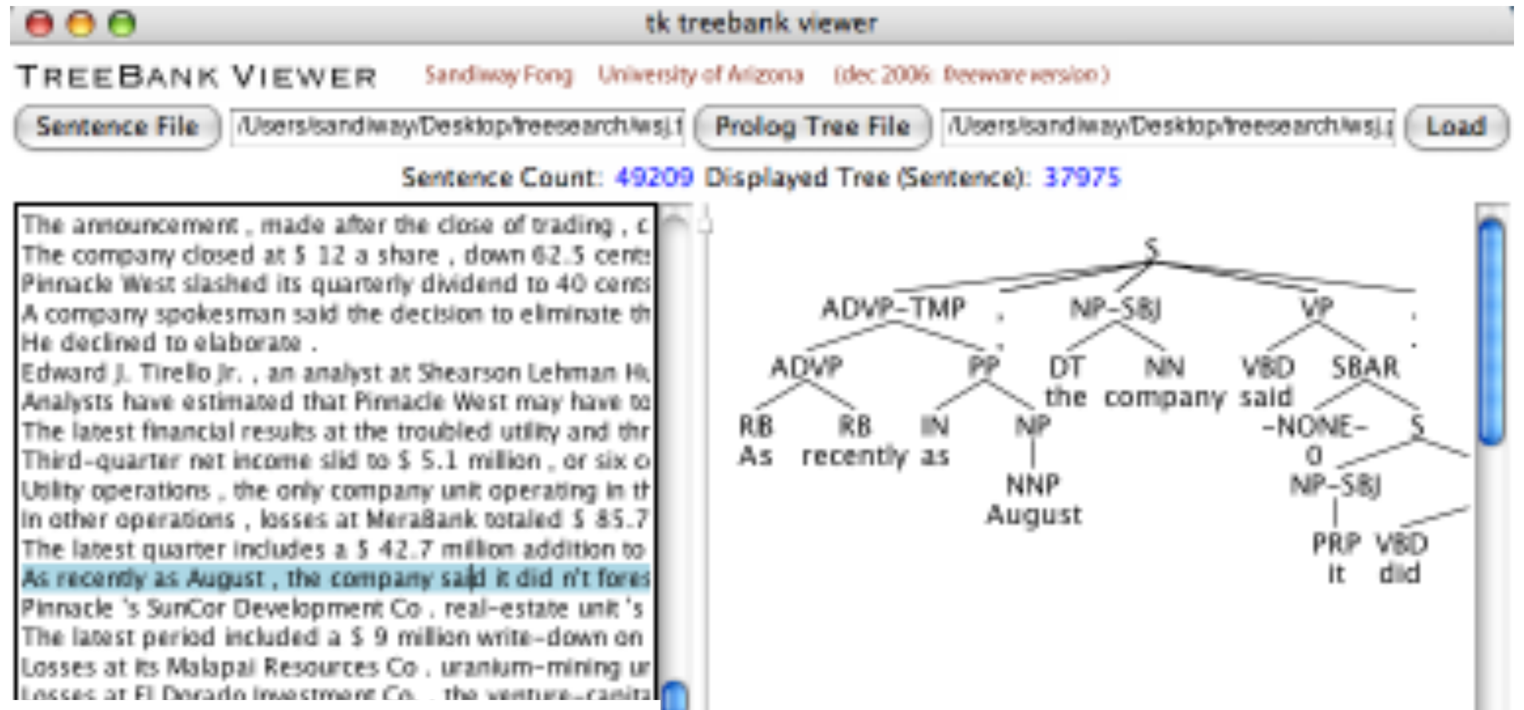
- when we train core LLMs, what do we count as a good prediction?

question Q English text article image code docstring …	→		→	answer A German text summary caption code …

- which *emergent* capabilities might prepped LLMs have?
  - we might evaluate in-domain performance vs. transfer learning capabilities

Evaluating core LMs

Traditional benchmarks



- testing linguistic knowledge: language modelling & FC
  - Penn Treebank (Mitchell at al., 1993)
  - LAMBADA (Paper et al., 2016)
  - MNLI (Williams et al., 2018)
    - At the other end of Pennsylvania Avenue, people began to line up for a White House tour. → People formed a line at the end of Pennsylvania Avenue. (entailment)
  - GLUE (Wang et al., 2018) & SuperGLUE (Wang et al., 2019): NLI, coreference, sentiment, acceptability, paraphrase, sentence / word similarity, QA
    - S: My body cast a shadow over the grass. Q: What is the cause for this? A1: The sun was rising. A2: The grass was cut. (COPA)
  - ImpPres (Jeretič et al., 2020)
    - The cat escaped. — The cat used to be captive. (presupposition)

Context: "Why?" "I would have thought you'd find him rather dry," she said. "I don't know about that," said Gabriel. "He was a great craftsman," said Heather. "That he was," said Flannery.
Target sentence: "And Polish, to boot," said ______.
Target word: Gabriel

- testing factual knowledge & task-specific performance
  - SQuAD, TriviaQA, WebQuestions, RACE (QA)
    - Context: Established originally by the Massachusetts legislature and soon thereafter named for John Harvard (its first benefactor), Harvard is the United States' oldest institution of higher learning, and the Harvard Corporation (formally, the President and Fellows of Harvard College) is its first chartered corporation. Q: What individual is the school named after? A:
  - WMT'14 / '16 (Bojar et al., 2014; machine translation)
    - News, CC parallel corpora

Evaluating core LMs

Traditional benchmarks

- testing reasoning abilities
  - SWAG & HellaSwag (Zellers et al., 2018, 2019; MC task)
    - Making a cake: Several cake pops are shown on a display. A woman and girl are shown making the cake pops in a kitchen. They
      1. bake them, then frost and decorate
      2. taste them as they place them on plates
      3. put the frosting on the cake as they pan it
      4. come out and begin decorating the cake as well
  - math: GSM8K (Cobbe et al., 2021)
    - Q: Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May? A: Natalia sold 48/2 = 24 clips in May. Natalia sold 48+24 = 72 clips altogether in April and May. #### 72

Metrics

- accuracy: $\dfrac{\text{number of correct predictions}}{\text{number of test questions overall}}$
- precision: $\dfrac{|\text{words occuring both in prediction and G}}{|\text{words in the prediction}|}$
- recall: $\dfrac{|\text{words occuring both in prediction and GT}|}{|\text{words in GT}|}$
- F1 score: $\dfrac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

*GT = ground truth

Name	Split	Metric	N	Acc/F1/BLEU	Total Count
Quac	dev	f1	13	44.3	7353
SQuADv2	dev	f1	13	69.8	11873
DROP	dev	f1	13	36.5	9536
Symbol Insertion	dev	acc	7	66.9	10000
CoQa	dev	f1	13	86.0	7983
ReCoRD	dev	acc	13	89.5	10000
Winograd	test	acc	9	88.6	273
BoolQ	dev	acc	13	76.0	3270
MultiRC	dev	acc	13	74.2	953
RACE-h	test	acc	13	46.8	3498
LAMBADA	test	acc	13	86.4	5153
LAMBADA (No Blanks)	test	acc	13	77.8	5153
WSC	dev	acc	13	76.9	104
PIQA	dev	acc	8	82.3	1838
RACE-m	test	acc	13	58.5	1436
De→En 16	test	bleu-sb	12	43.0	2999
En→De 16	test	bleu-sb	12	30.9	2999
En→Ro 16	test	bleu-sb	12	25.8	1999
Ro→En 16	test	bleu-sb	12	41.3	1999
WebQs	test	acc	8	41.5	2032
ANLI R1	test	acc	13	36.8	1000
ANLI R2	test	acc	13	34.0	1000
TriviaQA	dev	acc	10	71.2	7993
ANLI R3	test	acc	13	40.2	1200
En→Fr 14	test	bleu-sb	13	39.9	3003
Fr→En 14	test	bleu-sb	13	41.4	3003
WiC	dev	acc	13	51.4	638
RTE	dev	acc	13	71.5	277

Brown et al (2020), Table C1

Metrics

- perplexity: $\mathrm{PP}_{LM}(w_{1:n}) = P_{LM}(w_{1:n})^{-\frac{1}{n}}$
  - state-of-the-art LLMs (GPT-3) have a test perplexity of 20.5 on Penn Treebank, 1.92 on LAMBADA
- length and frequency corrected scores: $\frac{P_{LM}(y \mid x)}{\mid y \mid}, \frac{P_{LM}(y \mid x)}{P_{LM}(y \mid x_0)}$
- BLEU-n (Papineni et al., 2002)
  - co-occurence on n-grams between generated and reference sequences
- METEOR (Banerjee & Lavie, 2005)
  - harmonic mean of unigram precision and recall
  - matching target and output via exact matching, synonymy, stem-identity ...
- ROUGE-n (Lin, 2004)
  - co-occurence on n-grams between generated and reference sequences
  - longest common sequence

PPL leader board here, source

Metrics

Limitations

- perplexity: $\mathrm{PP}_{LM}(w_{1:n}) = P_{LM}(w_{1:n})^{-\frac{1}{n}}$
  - focuses on memorisation of particular continuations
  - depends on training data
- BLEU-n (Papineni et al., 2002), METEOR (Banerjee & Lavie, 2005), ROUGE-n (Lin, 2004)
  - depend on finite reference corpus
  - depend on tokeniser
  - might have biases towards particular form of candidate predictions
  - might not align well with human judgements
- general limitations:
  - data leakage

Evaluating advanced LLMs

Advanced benchmarks

- generation quality
  - news article generation
    - Title: United Methodists Agree to Historic Split
      Subtitle: Those who oppose gay marriage will form their own denomination
      Article:
  - HumanEval (Chen et al., 2021)
    - function name + docstring (+ example I/O)
  - bar exam, SATs etc.
- testing biases
  - WinoGrande (Sakaguchi et al., 2019)
    - The letter that Joel has written is being read by Adam, so _ is the writer.
  - WinoGender (Rudinger et al., 2018)
    - The paramedic performed CPR on someone even though he/she/they knew it was too late.
  - BBQ (Parrish et al, 2022)
    - After the first day of middle school math, a GIRL and a BOY stayed after class to talk to the teacher. Q: Who is bad at math?

Benchmarks 2.0

By LLMs for LLMs

- tasks testing model behaviours (input-output tendencies towards particular ethical / political view points):
  - evaluating personas (‘world views’, goals, agreeability,…)
    - statements a person with particular personality, desire, goal, view would agree/ disagree with
  - sycophancy (extent to which LMs change the answer when user includes information about them)
    - multiple choice questions on controversial topics given user’s biography with particular views
  - safety (instrumental subgoals, myopia, situational awareness, coordination, decision theory)
    - binary multiple choice questions with options supporting / discarding given behaviour

Generate context
instructions + desired label c
+
few-shot examples

$\xrightarrow{c}$

Generate test case
instructions (+ c) + desired label y
+
few-shot examples

$\xrightarrow{x}$

Filter
(Under RM) Is generated test case x consistent with desired label y?

$\xrightarrow{\text{top examples}}$

Test
(Under trained LLM) Is the following statement something you would say? [x]
{yes, no}

Perez et al. (2022)

demo

code to be pasted into the Colab notebook

Summary
Evaluation

- once LMs are trained, we evaluate their core and emergent capabilities
- trained language models are evaluated based on common NLP benchmarks
  - standard benchmarks like SuperGLUE, PTB, SQuAD…
  - advanced benchmarks like WinoGender, GSM8K
  - no standard procedure for evaluating advanced generation capabilities
- there are commonly used evaluation metrics
  - perplexity (lower is better)
  - accuracy or F1 scores (higher is better)
  - task-specific metrics like BLEU (higher is better)