Introduction to Large Language Models

Machine Learning Summer School
Arequipa 2025

Tal Linzen
New York University and Google

A little bit about me

- Only NLP (Natural Language Processing) speaker at this event (I think)!
- PhD from NYU in 2015
- Now:
    - Professor in the Department of Linguistics and the Center for Data Science
    - Part-time research scientist at Google
- Fun facts:
    - I attended MLSS in Cádiz in 2016
    - I learn Spanish in Peru

What I Do: How Cognitive Science Can Help Us Understand Large Language Models, and Vice Versa

Targeted evaluation for generalization

Is language models' use of language consistent with the rules of grammar and logic?

Mimicking human language acquisition

Towards sample-efficient language learning in developmentally plausible settings

Analysis and interpretability

What architectural features and internal representations explain the models' successes and failures?

Prediction in humans and LMs

Towards LMs that make similar next-word predictions as humans

What kinds of language technologies would we like to have (“NLP tasks”)?

Answering questions based on a text

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain in scattered locations are called "showers".

What causes precipitation to fall?
gravity

What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?
graupel

Where do water droplets collide with ice crystals to form precipitation?
within a cloud

(Rajpurkar et al 2016)

Summarization

Source Document

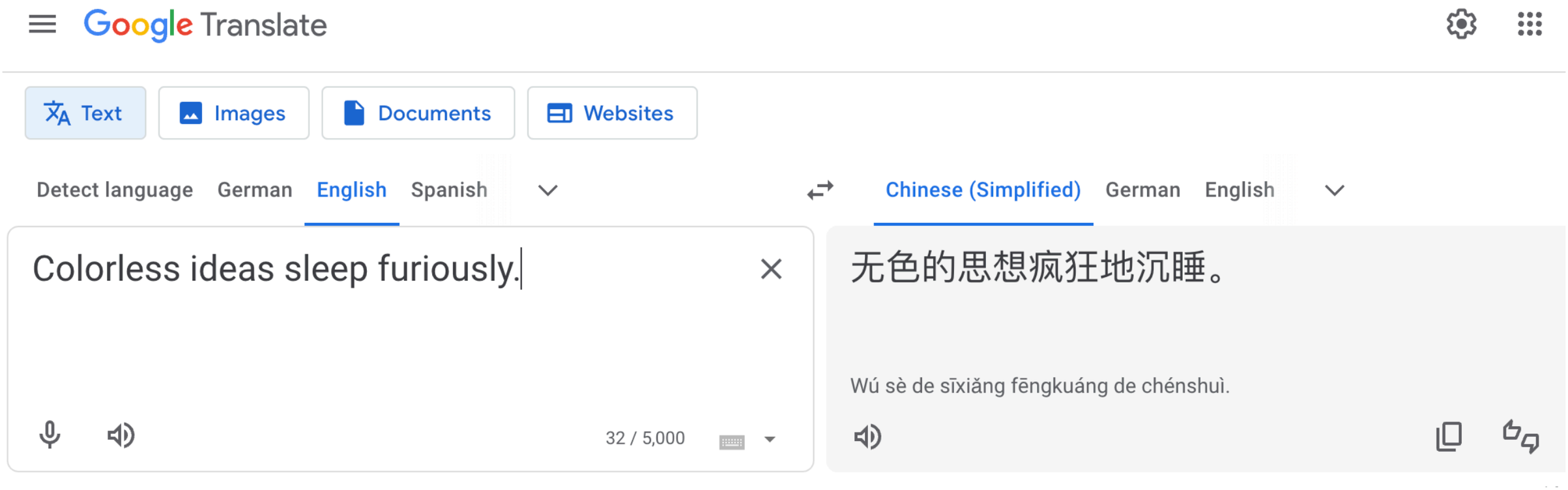
german chancellor angela merkel [did] not [look] too pleased about the weather during her [annual] easter holiday [in italy.] as britain [basks] in [sunshine] and temperatures of up to 21c, mrs merkel and her husband[, chemistry professor joachim sauer,] had to settle for a measly 12 degrees. the chancellor and her [spouse] have been spending easter on the small island of ischia, near naples in the mediterranean for over a [decade.]
[not so sunny:] angela merkel [and] her husband[, chemistry professor joachim sauer,] are spotted on their [annual] easter trip to the island of ischia[,] near naples[. the] couple [traditionally] spend their holiday at the five-star miramare spa hotel on the south of the island [, which comes] with its own private beach [, and balconies overlooking the] ocean [.]...

Reference

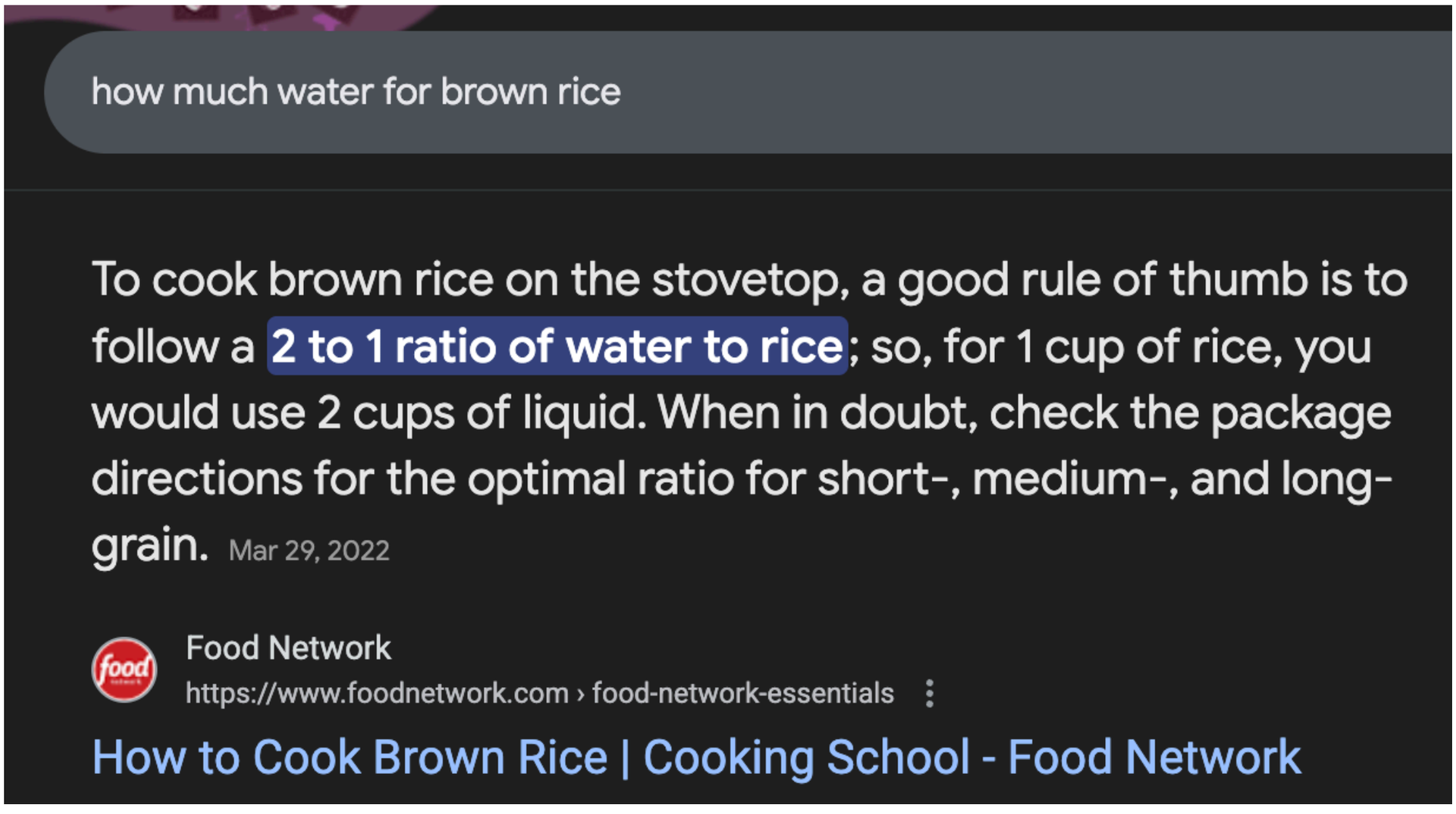
- angela merkel and husband spotted while on italian island holiday.

(Gehrman et al 2018)

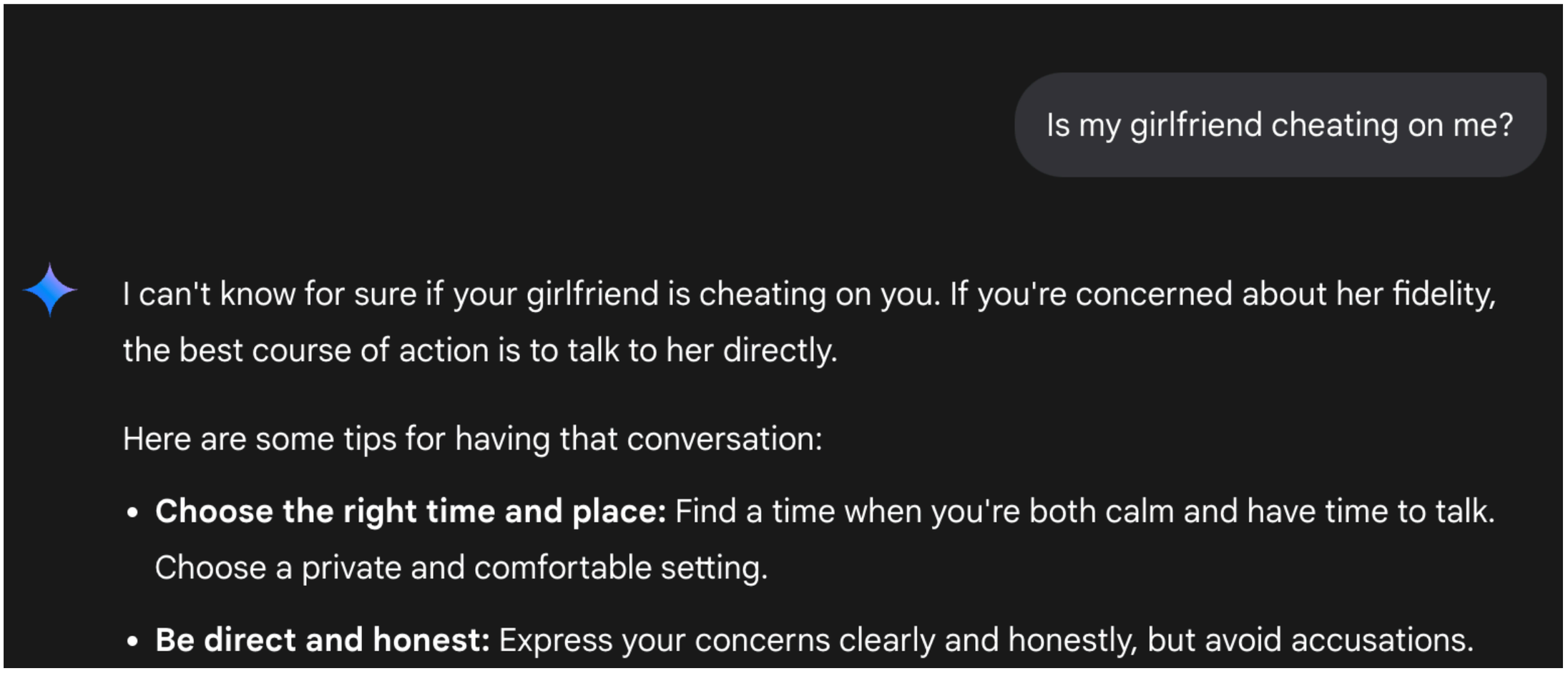
Machine translation



Search and information extraction



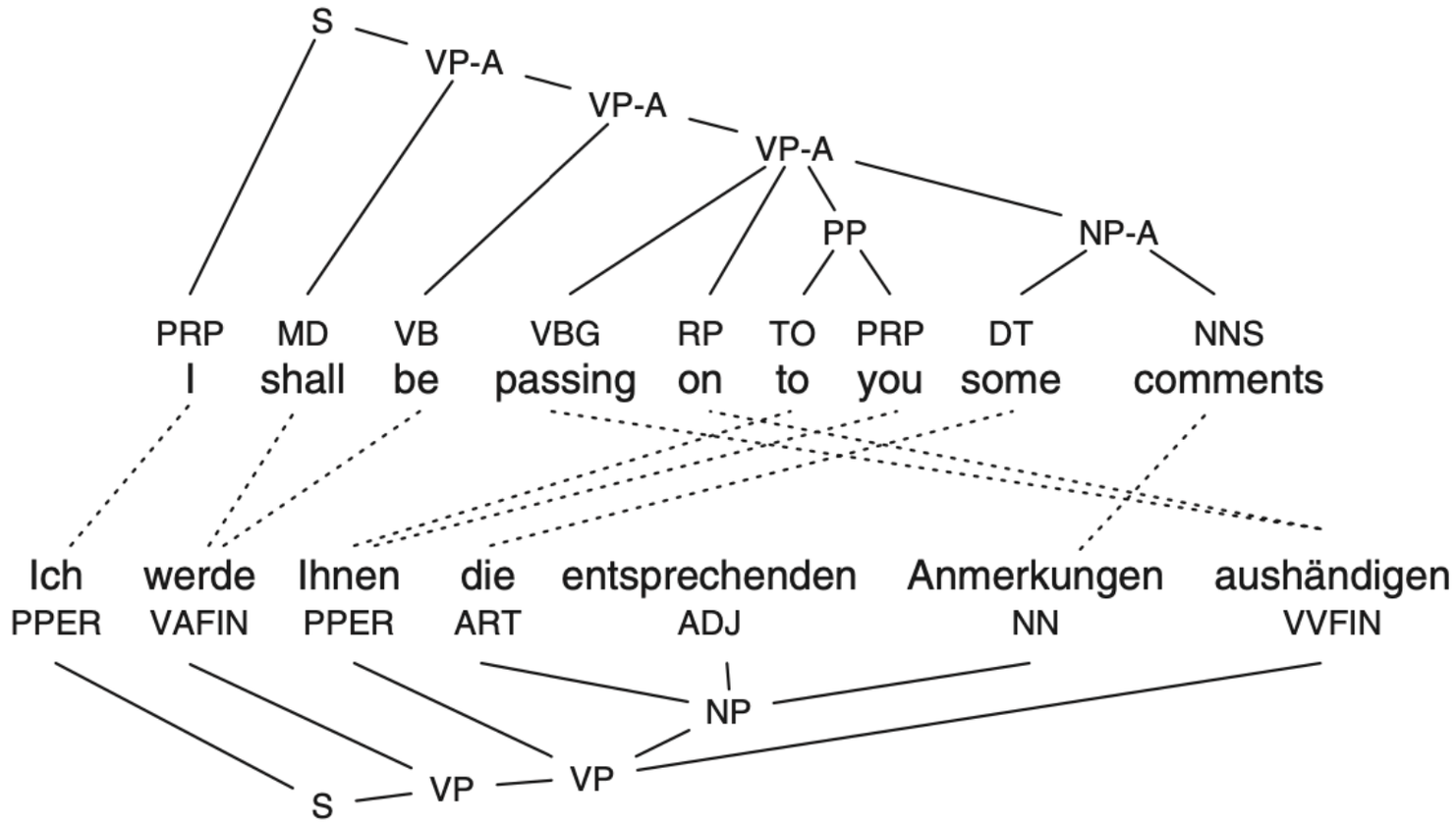
More ambitious AI assistants (e.g. for mental health applications)



Very brief history of NLP paradigms

Rule-based symbolic NLP

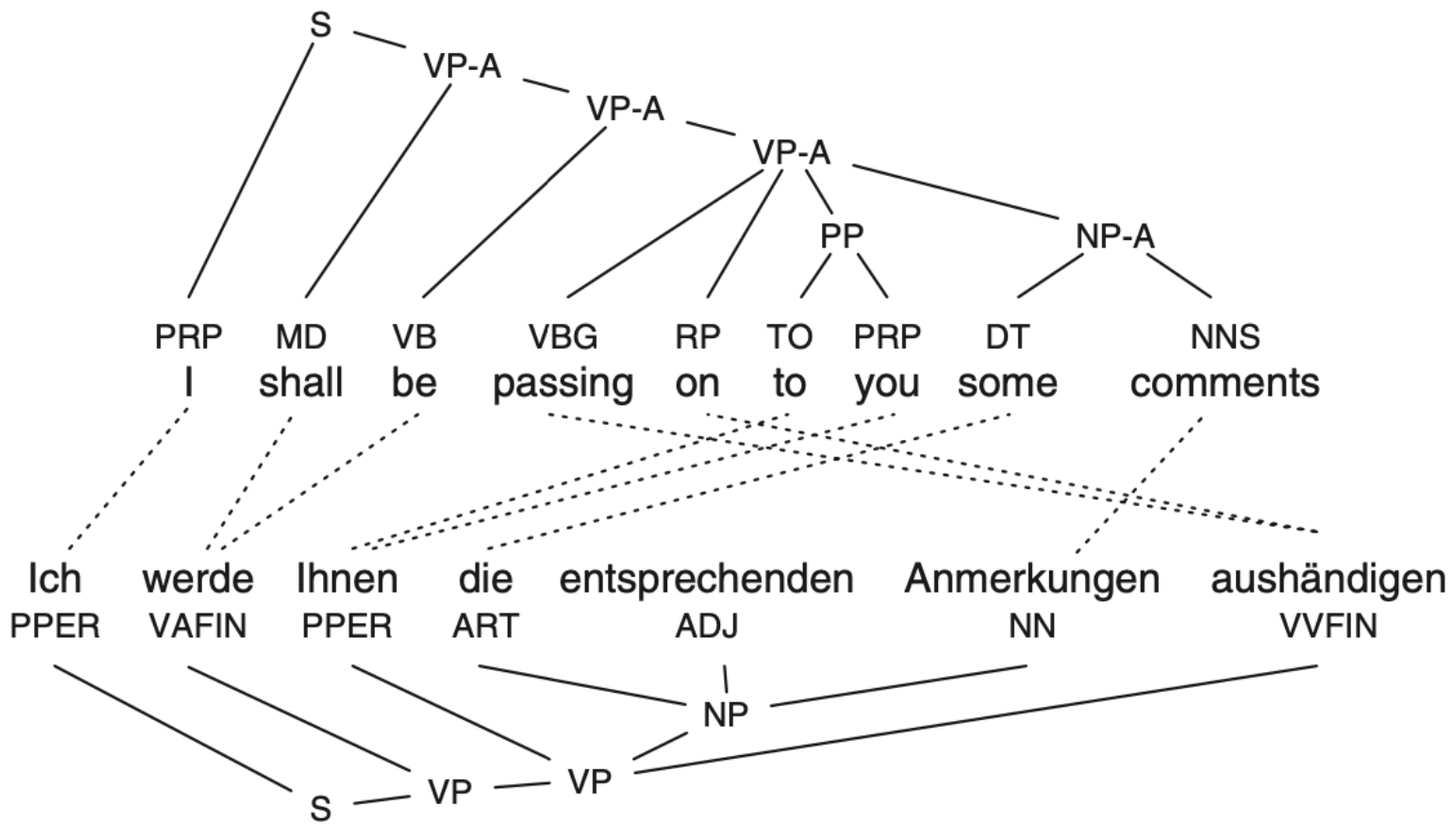
To translate from English to German, move the verb to the end (in certain cases)



(Koehn, 2010)

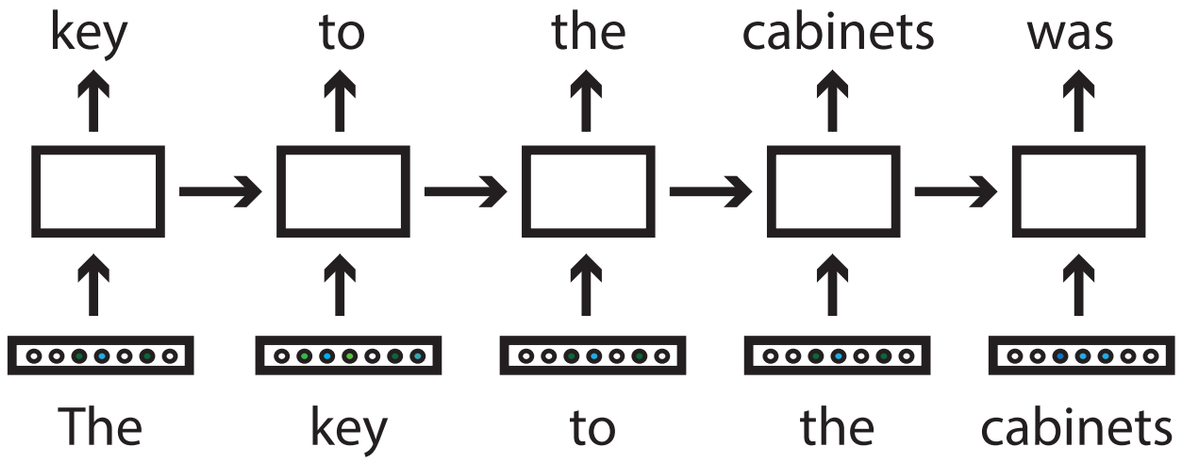
ML-based symbolic NLP

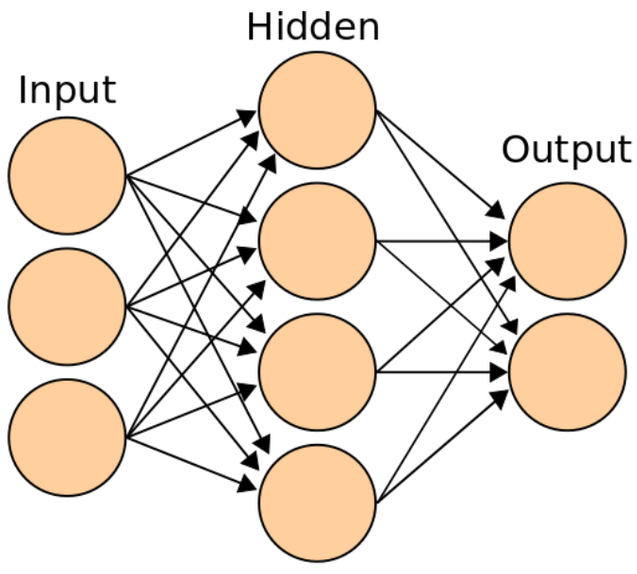
Based on examples of parallel sentences, figure out on your own that the verb needs to be moved

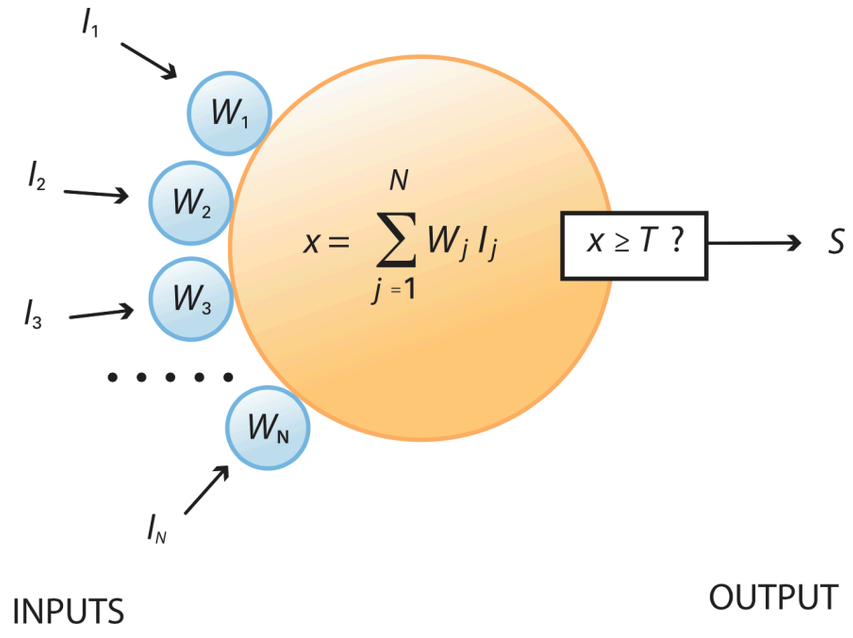


(Koehn, 2010)

Neural NLP: representation learning replace symbolic representations







Neural NLP: learning representations from examples

My dear Athos, we are enveloped in a network of spies.

Mon cher Athos, nous sommes enveloppés dans un réseau d'espions!

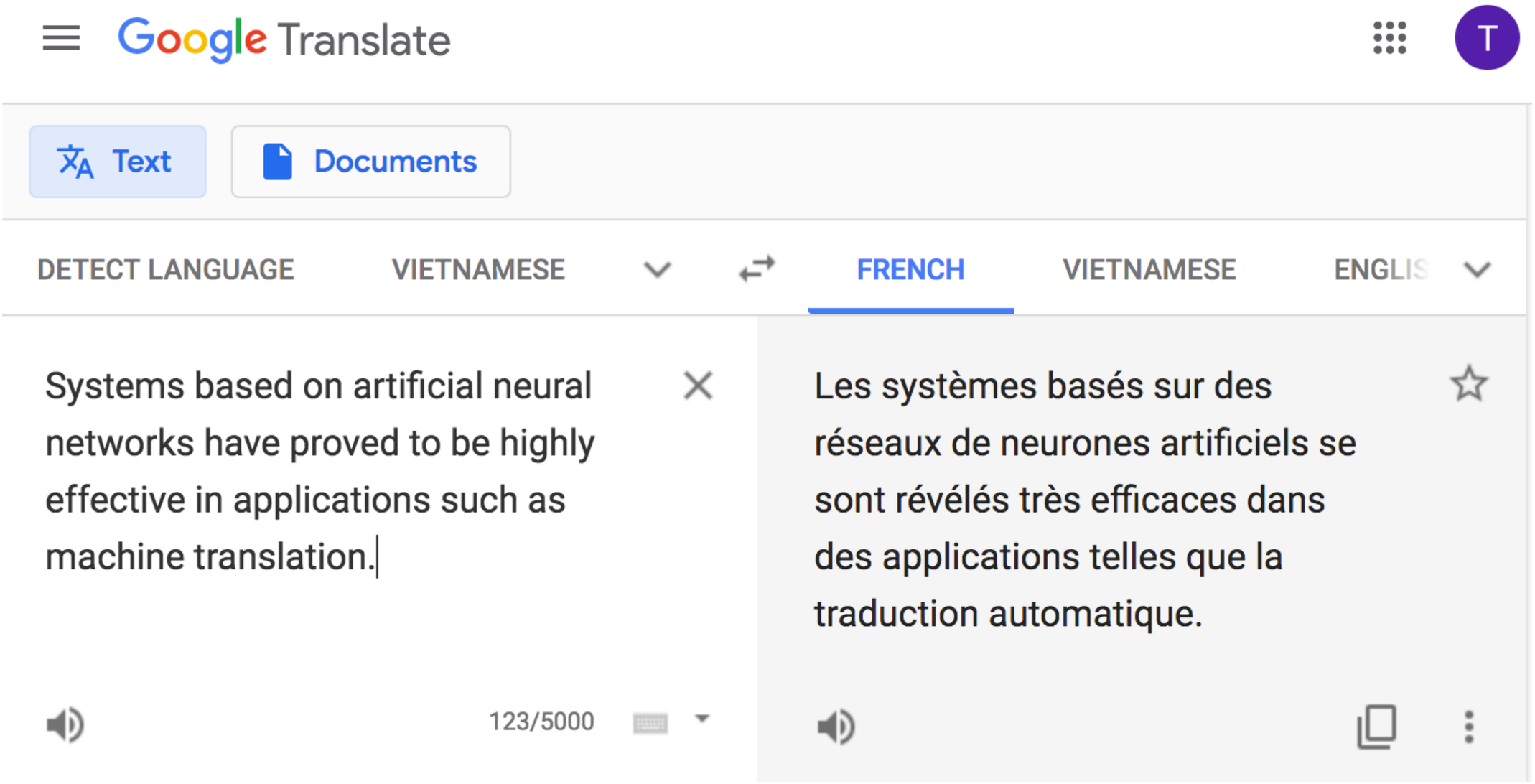
These two great thoroughfares, intersected by the two first, formed the canvas upon which reposed, knotted and crowded together on every hand, the labyrinthine network of the streets of Paris.

Ces deux grandes voies, croisées avec les deux premières, formaient le canevas sur lequel reposait, noué and serré en tous sense, le réseau dédaléen des rues de Paris.

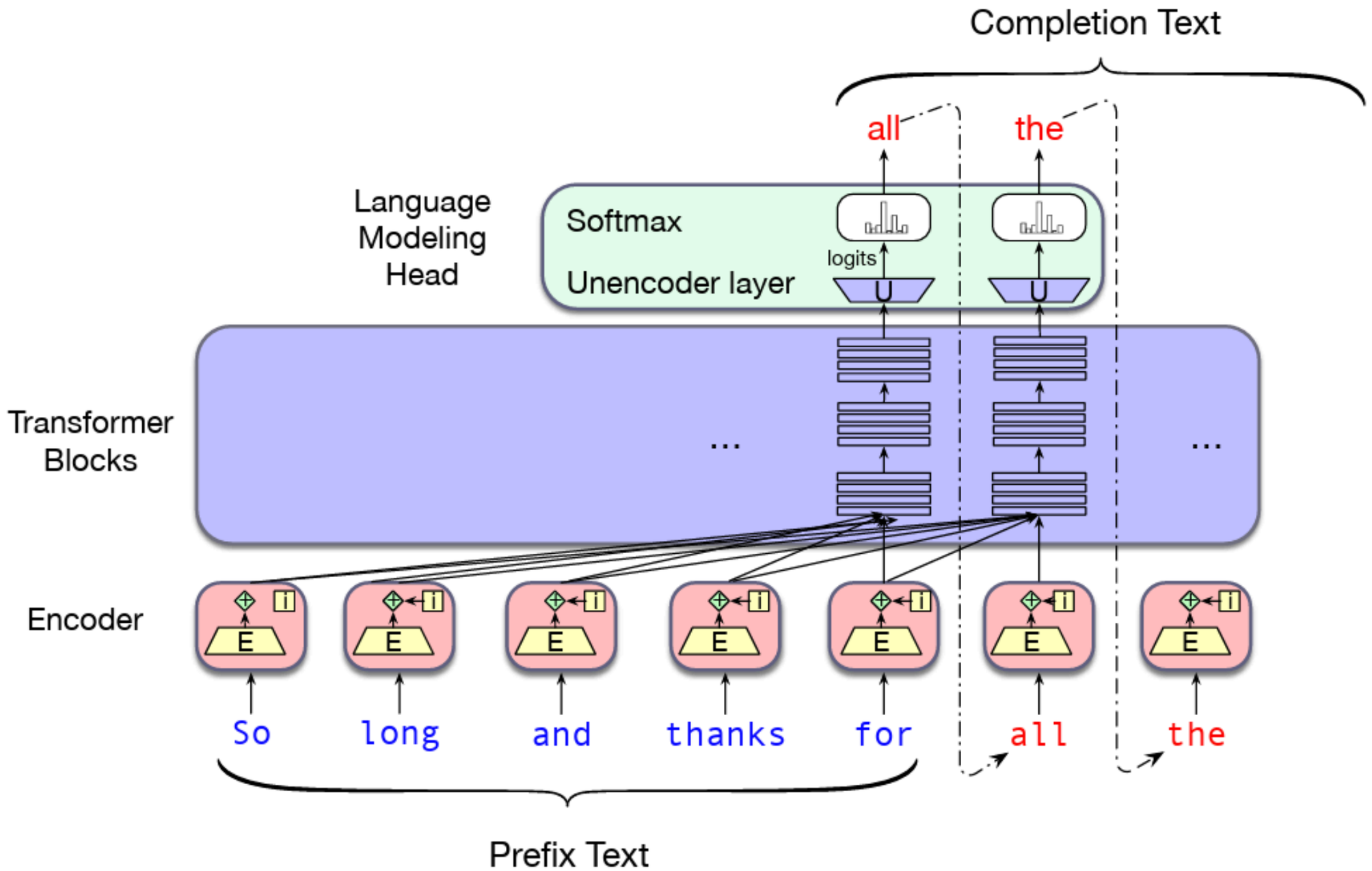
She flies, she is joyous, she is just born; she seeks the spring, the open air, liberty: oh, yes! But let her come in contact with the fatal network, and the spider issues from it, the hideous spider!

Elle vole, elle est joyeuse, elle vient de naître; elle cherche le printemps, le grand air, la liberté; oh! Oui, mais qu'elle se heurte à la rosace fatale, l'araignée en sort, l'araignée hideuse!

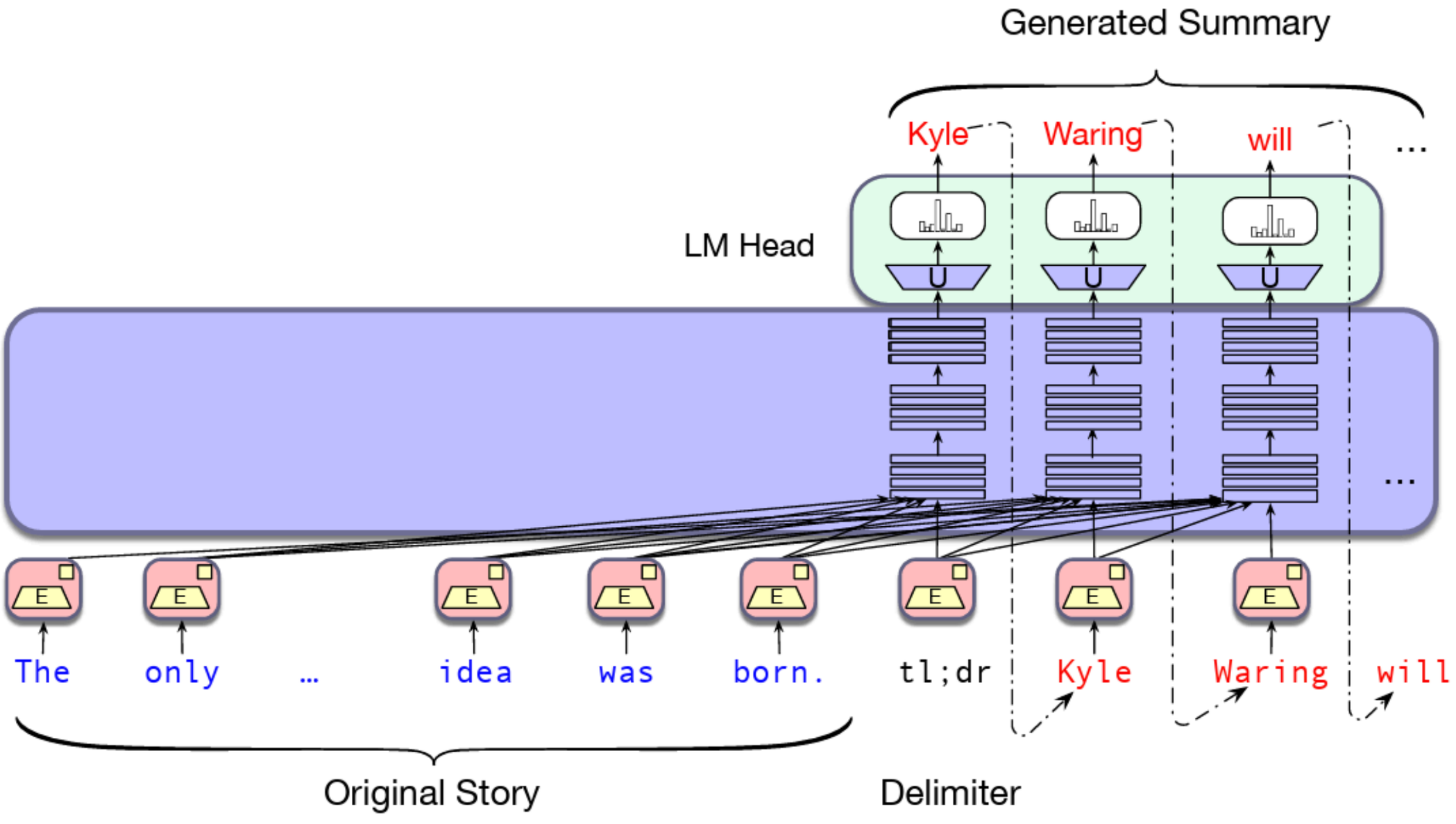
Neural NLP: learning representations from examples



The LLM paradigm: pre-training



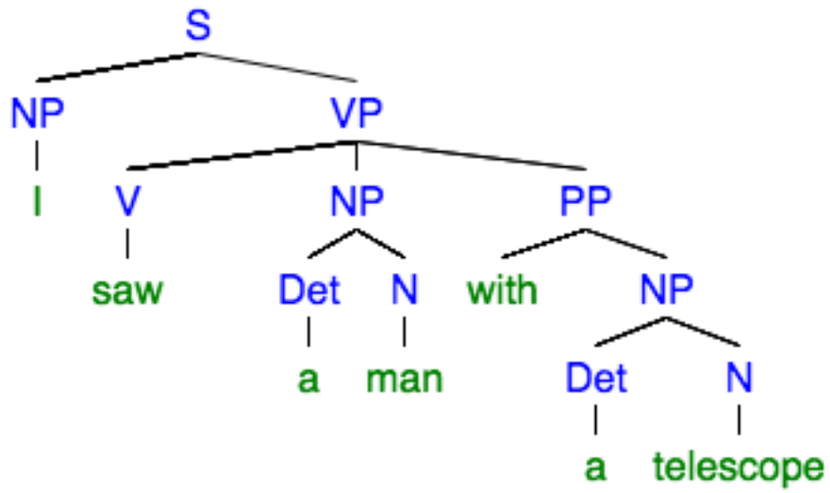
The LLM paradigm: aspirationally performing tasks zero-shot or few-shot

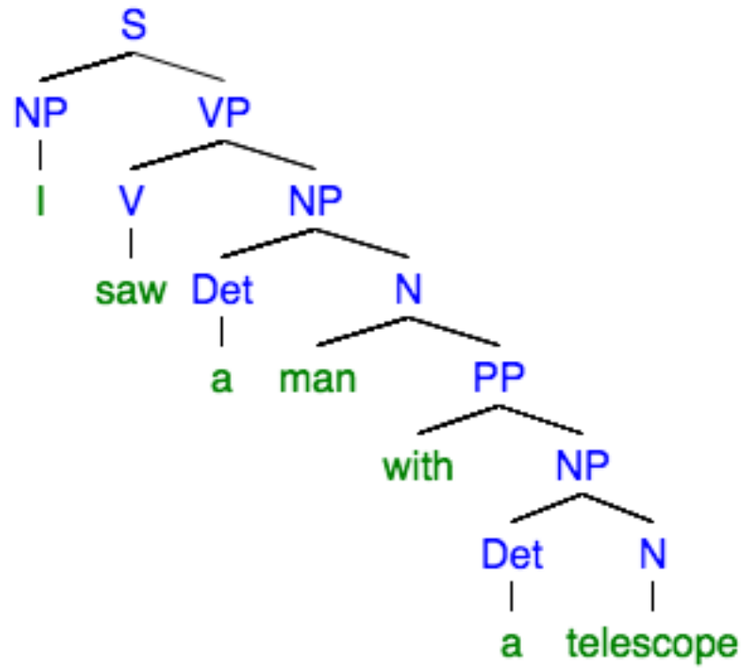


Why is language processing hard?

Ambiguity

I saw a man with a telescope





The interpretation can often be disambiguated by *context* (previous sentences) or *world knowledge* (facts about the world that are not mentioned). E.g. this has only one interpretation:

I ate soup with noodles

Pragmatics

- The intended meaning is often different from the literal one

- The listener needs to understand not only **what** the speaker is saying, but also **why** they are saying it

Linguistic variation

- A lot of research focuses on English, but the majority of language technology users speak other languages

- Even English is not a single language either: there are many different *dialects* of English, in the United States and in other countries (UK, India, Australia, Nigeria)

- Other languages are much more diverse than English (e.g. regional varieties of Italian)

Challenges of linguistic diversity: ambiguity in word segmentation

A sentence in Chinese	下雨天留客天留我不留
Interpretation 1	下雨，天留客。天留，我不留！
Interpretation 2	下雨天，留客天。留我不？留！

Figure 1
A Chinese sentence with ambiguity of phrasing.

A sentence in Chinese	我喜欢新西兰花
Interpretation 1	我 喜欢 新西兰 花
Interpretation 2	我 喜欢 新 西兰花

Figure 2
An example that can be segmented in two different ways.

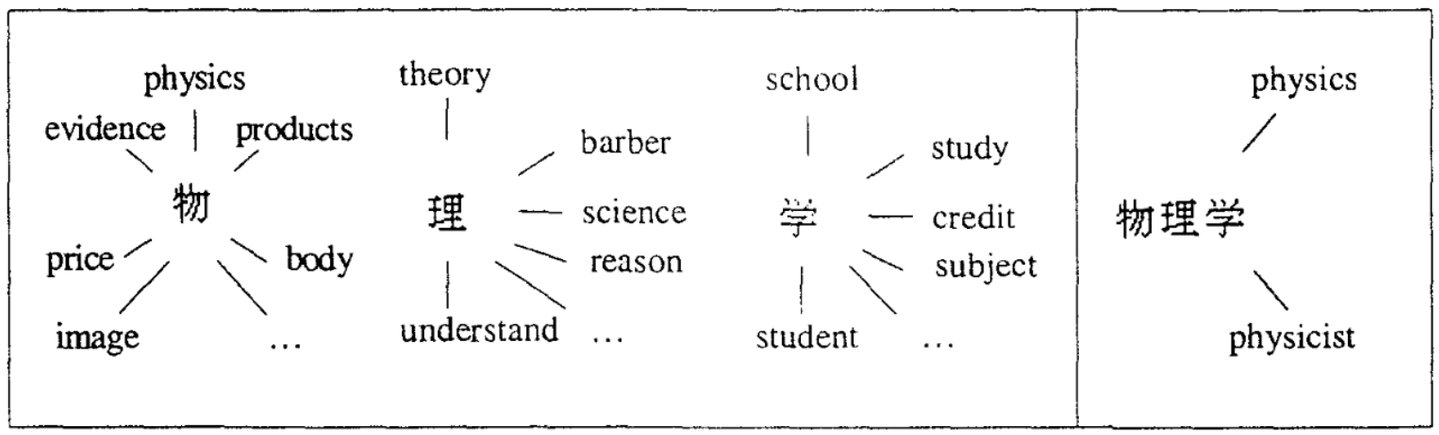


Figure 3
Example of treating each character in a query as a word.

(Teahan et al., 2000)

Syntactic differences across languages

- What is the order of the subject, the verb and the object?

English: *He wrote a letter to a friend*

Japanese: *tomodachi ni tegami-o kaita*
friend to letter wrote

Morphological diversity

- Morphology is (roughly) the internal grammar of words: how we construct words from smaller units, called *morphemes:*
  - *penguins = penguin + s*
  - *parallelize = parallel + ize*

Morphological diversity

- Languages differ enormously in their morphological systems

- E.g., Turkish is “highly agglutinative”; this entails that, unlike in English, many of the words we will come across are words we have never seen before

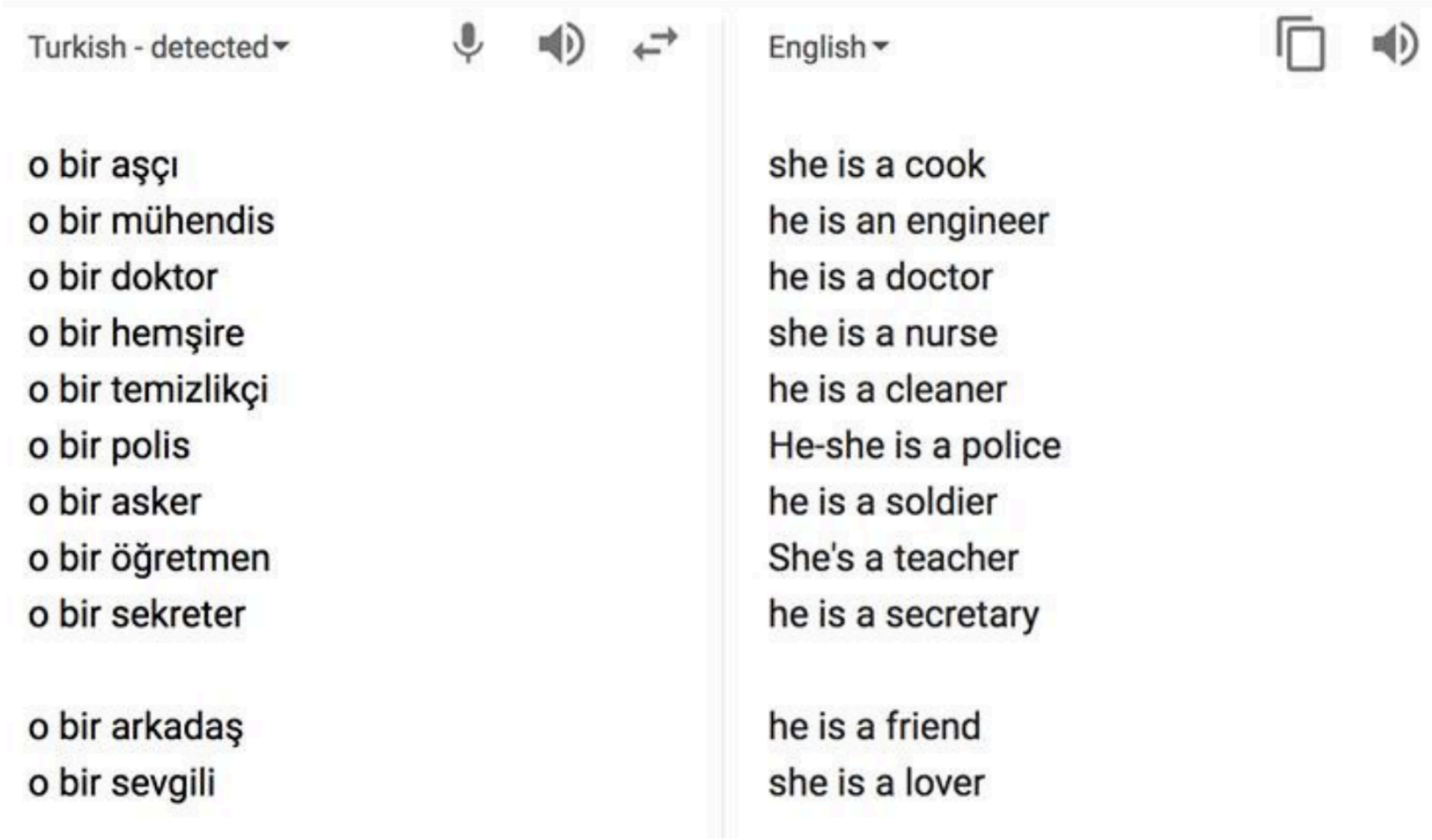
I will be able to go.

(go) + (able to) + (will) + (I)

git + ebil + ecek + im

Gidebileceğim.

Challenges of linguistic diversity: gender marked pronouns



Language models: probability distributions over sequences of words

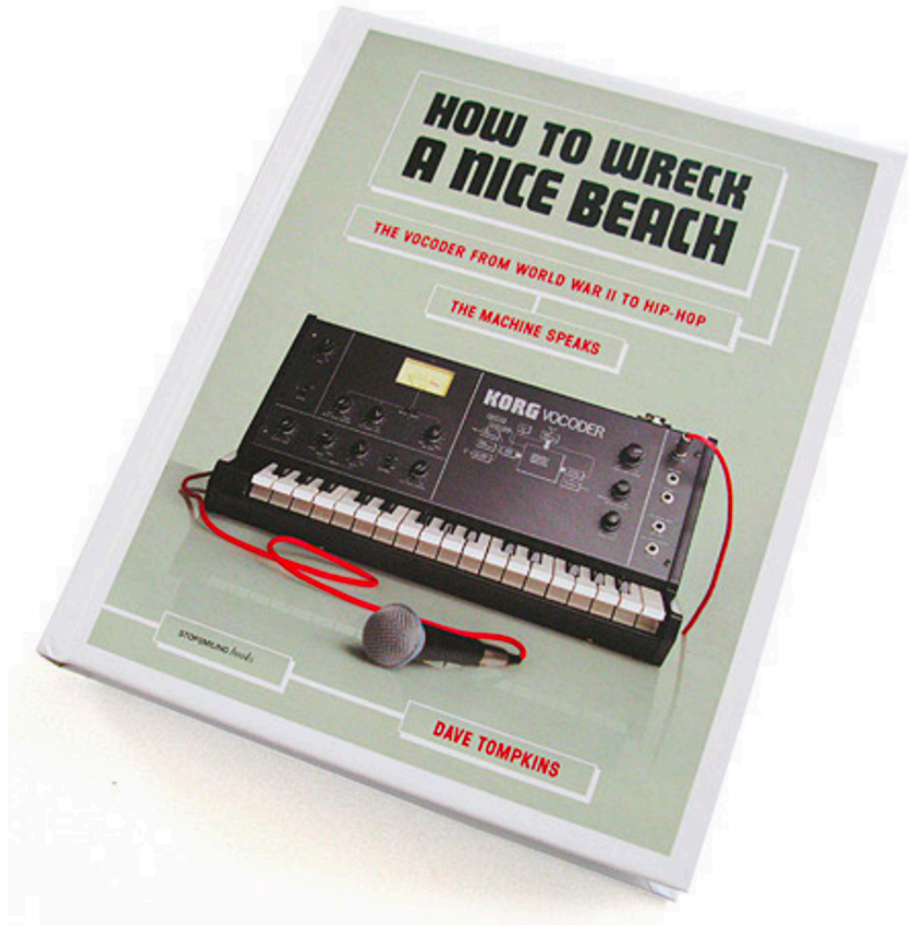
Probabilistic language models

- A language model defines a probability distribution over all possible sequences of words
- $P(s) = P(w_1 w_2 \ldots w_{n_s})$
- $\sum_s P(s) = 1$
- We're using this as a technical definition: a "language model" is not just any probabilistic model used in language technologies

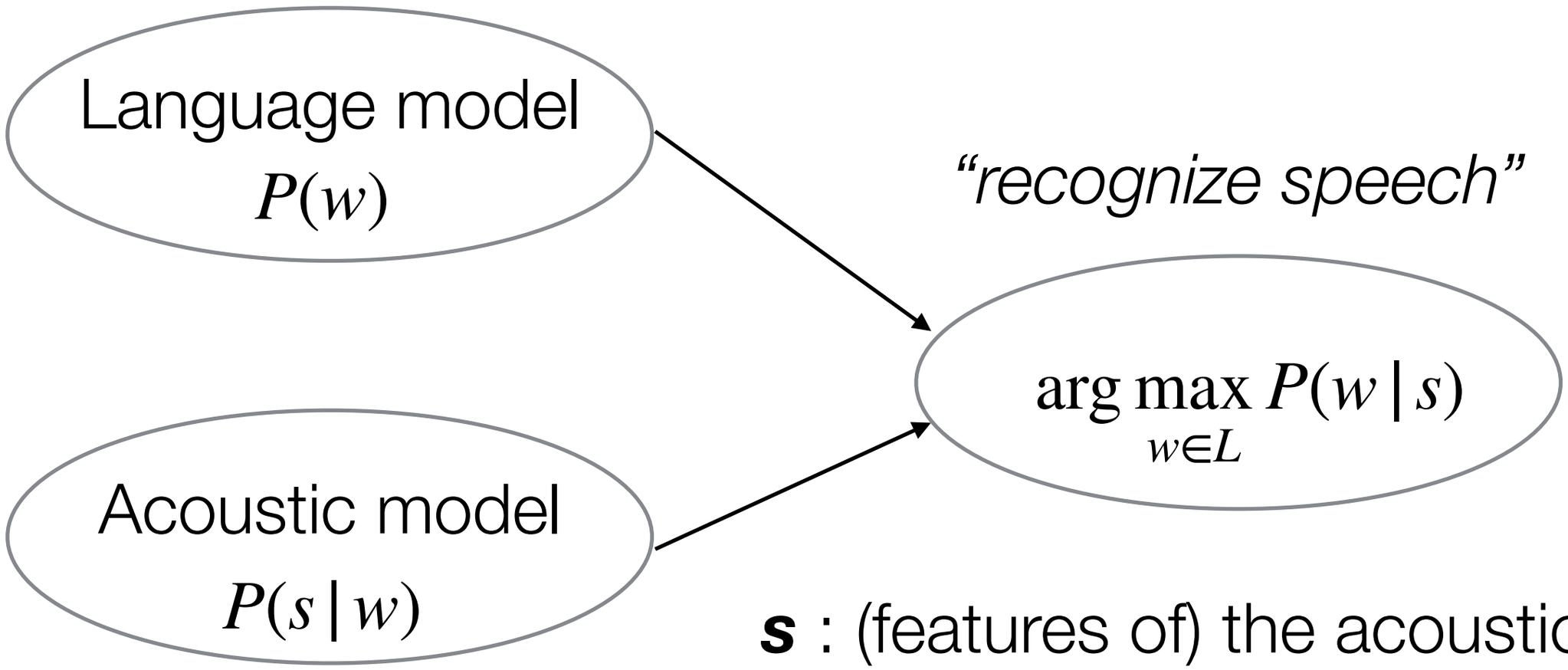
Classic applications of language models: speech recognition

- “how to recognize speech” and “how to wreck a nice beach” sound very similar

- If you know nothing about the context, which one would you say if more likely?



Using language models for automatic speech recognition



s : (features of) the acoustic signal

w : a word

L : the lexicon (set of all words)

Probabilistic language models

- How do we compute the probability of the sequence *its water is so transparent that*?
  - P(*its*, *water*, *is*, *so*, *transparent*, *that*)
- Chain rule of probability: $P(A, B) = P(A)P(B \mid A)$
- Applied again: $P(A, B, C) = P(A)P(B \mid A)P(C \mid A, B)$
- We use the chain rule to decompose P(*its*, *water*, *is*, *so*, *transparent*, *that*)

Probabilistic language models

- We're estimating P(*its*, *water*, *is*, *so*, *transparent*, *that*)
- How can we estimate P(*water* | *its*) from a corpus (a large collection of texts)?
- What about P(*is* | *its*, *water*)?
- And P(*that* | *its, water, is, so, transparent*)?
- What kind of simplifying assumption can we make?

The Markov assumption

- Assume that P(*that* | *its, water, is, so, transparent*) ≈ P(*that* | *transparent*): a bigram model
- Or P(*that* | *its, water, is, so, transparent*) ≈ P(*that* | *so, transparent*): a trigram model
- An *n*-gram model conditions only on the last *n*-1 words
- Is this in general a reasonable assumption about language?
- No: The **key** that I left on the table by the armchairs **is**
- But this assumption can nevertheless be effective in many cases

n-gram models

Unigram: $$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i)$$

Bigram: $$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-1})$$

Terminology points: an **n**-gram model conditions on the last **n-1** words

Corpus counts

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

(Figure from Jurafsky & Martin “Speech and Language Processing, like many others in these slides)

Language model evaluation: sampling from the model

1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

(Trained on the Wall Street Journal corpus)

Perplexity

- Inverse probability assigned to a held-out test set, normalized by the number of words:

$$PP(W) = P(w_1 w_2 ... w_N)^{-\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1 w_2 ... w_N)}}$$

$$\log PP(W) = -\frac{\log P(w_1 w_2 \ldots w_N)}{N}$$

On the WSJ:

	Unigram	Bigram	Trigram
Perplexity	962	170	109

Going beyond n-grams with neural network language models

Two sparsity issues with n-gram models

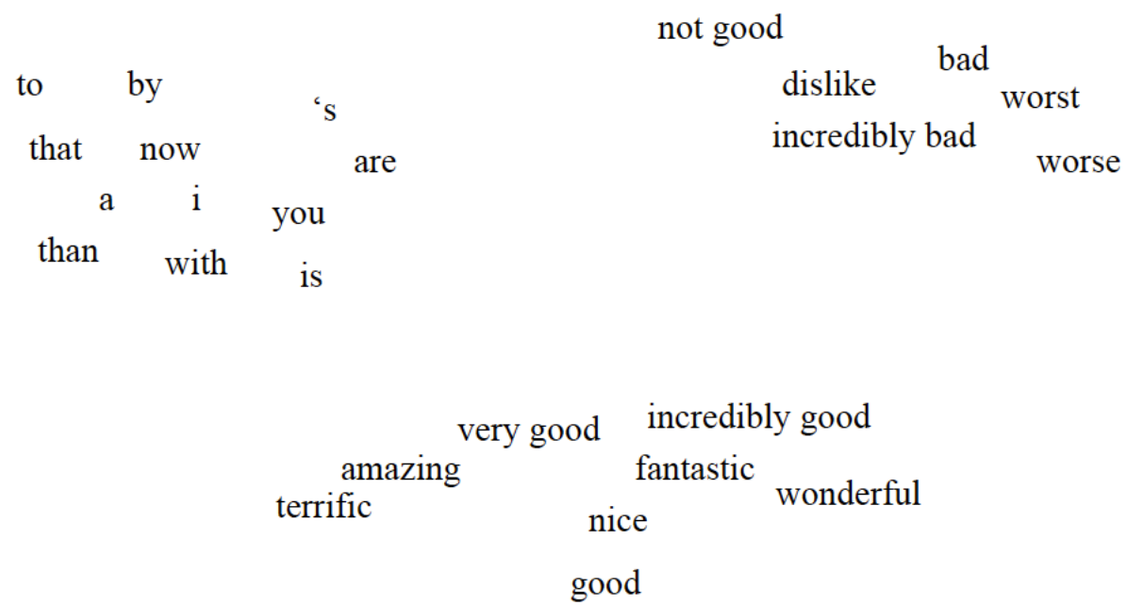
- Issue 1: no similarity-based generalization:
  - *I enjoy eating avocados* (occurs in the training corpus)
  - *I enjoy eating pupusas* (not in the training corpus, should still be assigned some probability)
  - *I enjoy eating democracy* (not in the training corpus, should be assigned zero or a very low probability)

Two sparsity issues with n-gram models

- Issue 2: Limited context window (only last n-1 words)
- Can’t capture statistical dependencies such as subject-verb agreement:
- The **key** to the cabinets by the door **is**…

Capturing similarity with word vector representations (word embeddings)

bad: (1, -0.5, 3, 2)
worse: (1.2, -0.6, 3.2, 2)
with: (-3, 4.3, 0.5, -1)



(PCA, tSNE)

Coming up with these vectors: distributional semantics

... lady to travel by daxoople, though she was perfectly...
... get married on an daxoople and then jump out...
... and nightmares of an daxoople about to crash. But...
... to level out the daxoople, decreasing the ground clutter...
... real cars from the daxoople to set up this...
... flights. The air in daxoople cabins has very low humidity...
... Carnival trip, the paper daxoople contest was a lot...
... soar. We used an daxoople as an example for...

Coming up with these vectors: distributional semantics

... lady to travel by airplane, though she was perfectly...
... get married on an airplane and then jump out...
... and nightmares of an airplane about to crash. But...
... to level out the airplane, decreasing the ground clutter...
... real cars from the airplane to set up this...
... flights. The air in airplane cabins has very low humidity...
... Carnival trip, the paper airplane contest was a lot...
... soar. We used an airplane as an example for...

Similarity-based generalization with neural language modeling

The boys went outside to _____

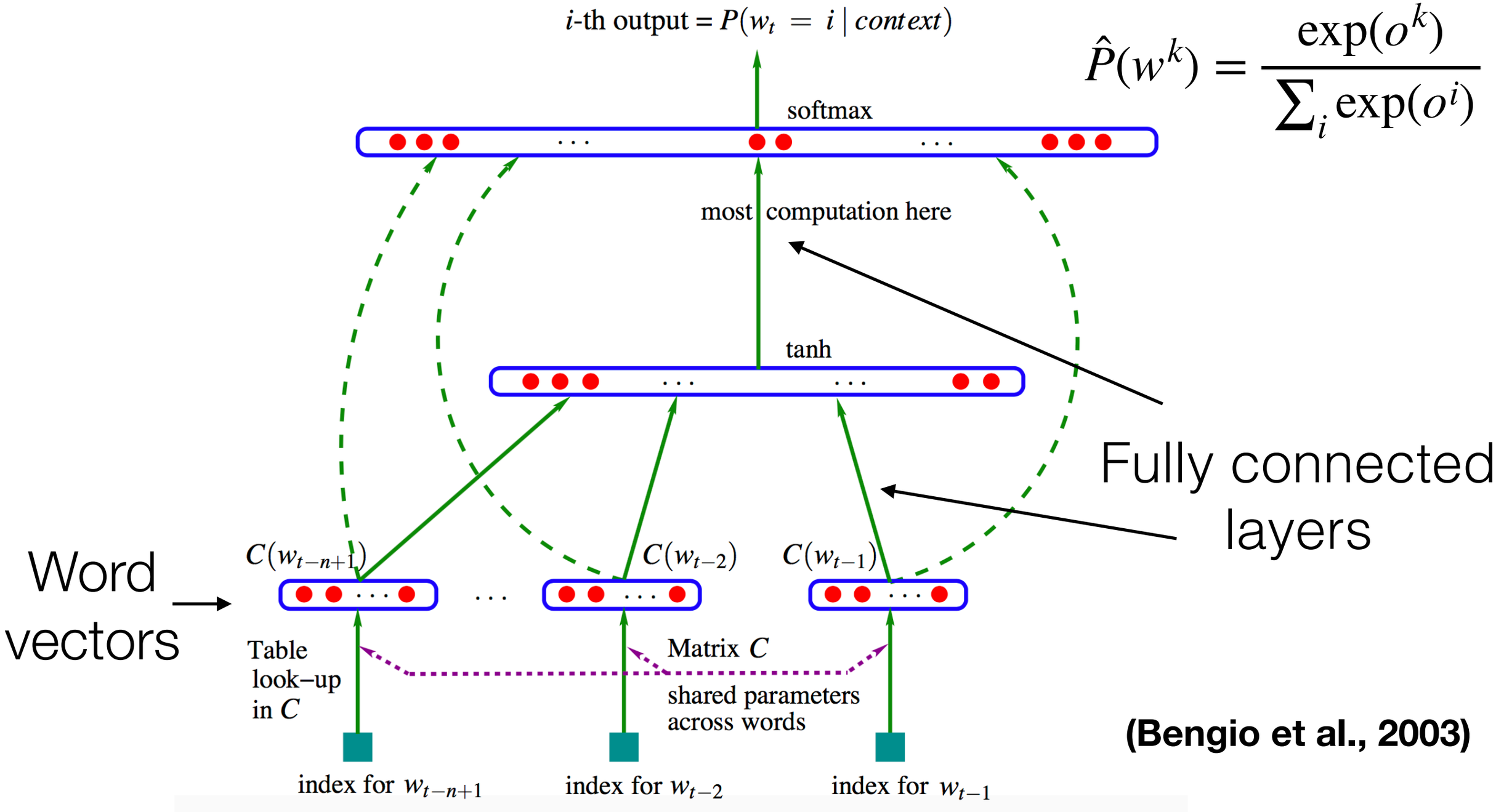
$$\hat{P}(w_n = w^k \mid w_1, \ldots, w_{n-1})$$

Objective function: minimize the surprisal of the word that in fact occurred in the corpus (“cross entropy loss”):

$$-log\hat{P}(w)$$

Similarity-based generalization with neural language modeling

$$\hat{P}(w^k) = \frac{\exp(o^k)}{\sum_i \exp(o^i)}$$



(Bengio et al., 2003)

Context is last n - 1 words!

An aside on tokenization

Tokenization: splitting the input into units

- The natural place to split the input appears to be spaces between words

- But we keep coming across new words!

- E.g. morphology is productive, and we would like to have representations for the following words:

  - Vectorize

  - LSTMification

  - Goldbergian

The problem is more severe in e.g. Turkish

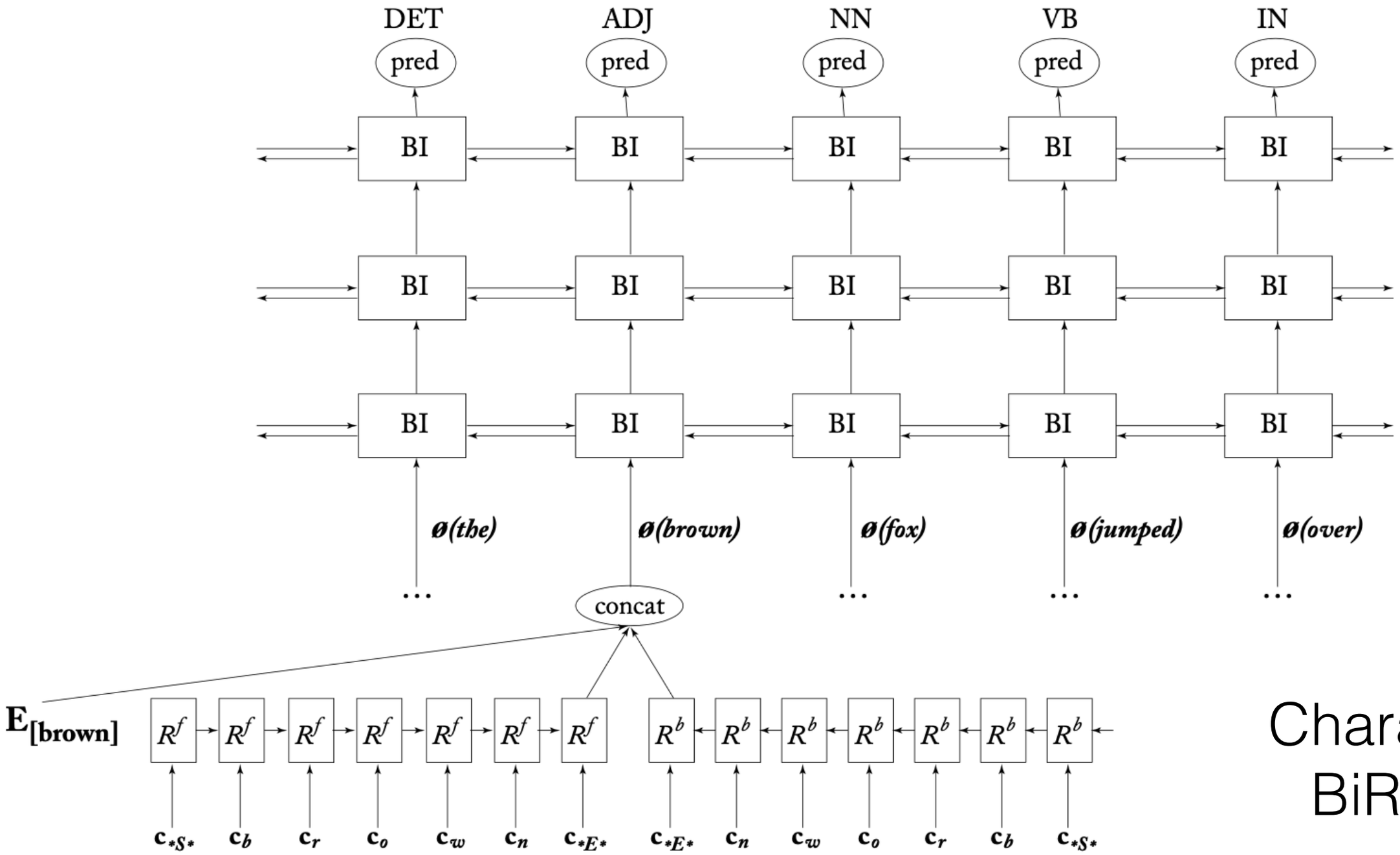
I will be able to go.

(go) + (able to) + (will) + (I)

git + ebil + ecek + im

Gidebileceğim.

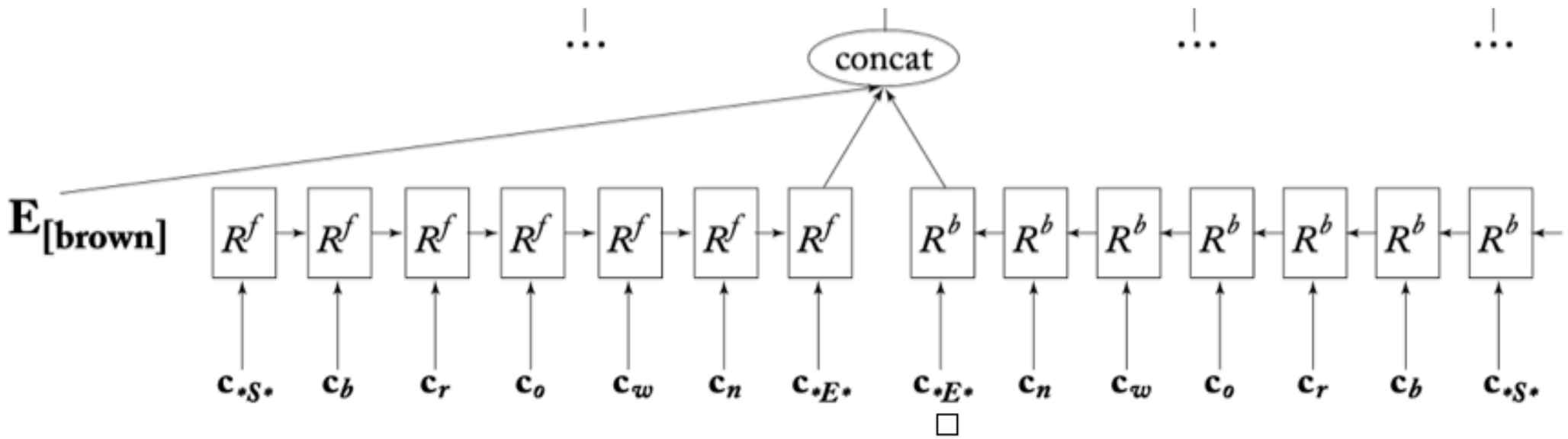
Character-based word embeddings



Character BiRNN

Character-based word embeddings

The embedding for each word is generated dynamically



We still train a full word embedding (useful for frequent words)

Character embeddings

Byte-pair encoding (BPE)

Starting from a character-only vocabulary, we learn the vocabulary by progressively merging frequent bigrams:

corpus	**vocabulary**
5 l o w _	_, d, e, i, l, n, o, r, s, t, w
2 l o w e s t _	
6 n e w e r _	
3 w i d e r _	
2 n e w _	

corpus	**vocabulary**
5 l o w _	_, d, e, i, l, n, o, r, s, t, w, er
2 l o w e s t _	
6 n e w er _	
3 w i d er _	
2 n e w _	

Byte-pair encoding (BPE)

corpus	**vocabulary**
5 l o w _	_, d, e, i, l, n, o, r, s, t, w, er
2 l o w e s t _	
6 n e w er _	
3 w i d er _	
2 n e w _	

corpus	**vocabulary**
5 l o w _	_, d, e, i, l, n, o, r, s, t, w, er, er_
2 l o w e s t _	
6 n e w er_	
3 w i d er_	
2 n e w _	

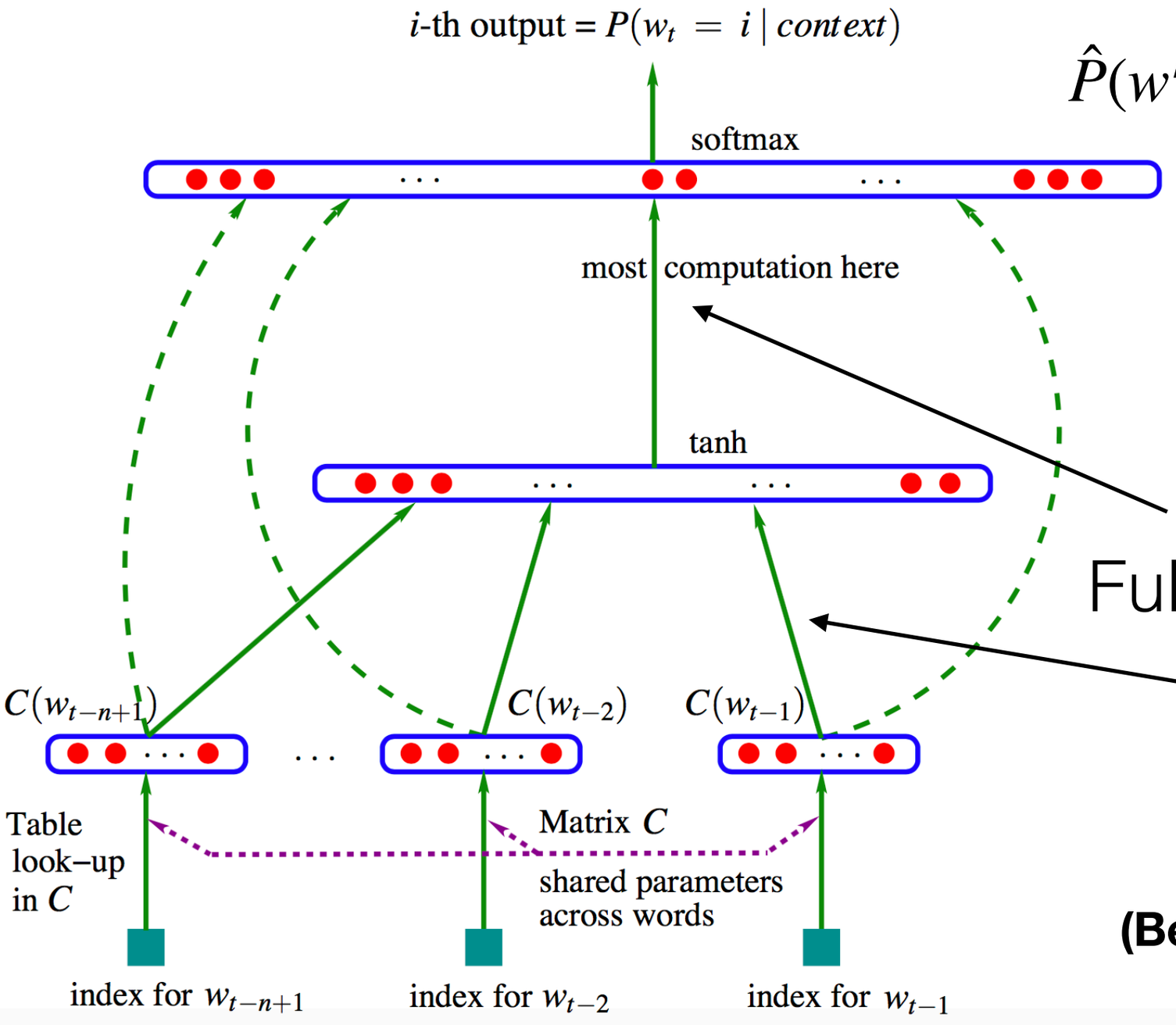
Byte-pair encoding (BPE)

Merge	**Current Vocabulary**
(ne, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new
(l, o)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo
(lo, w)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low
(new, er_)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_
(low, _)	_, d, e, i, l, n, o, r, s, t, w, er, er_, ne, new, lo, low, newer_, low_

Language models with an extended context window

Similarity-based generalization with neural language modeling

$$\hat{P}(w^k) = \frac{\exp(o^k)}{\sum_i \exp(o^i)}$$



(Bengio et al., 2003)

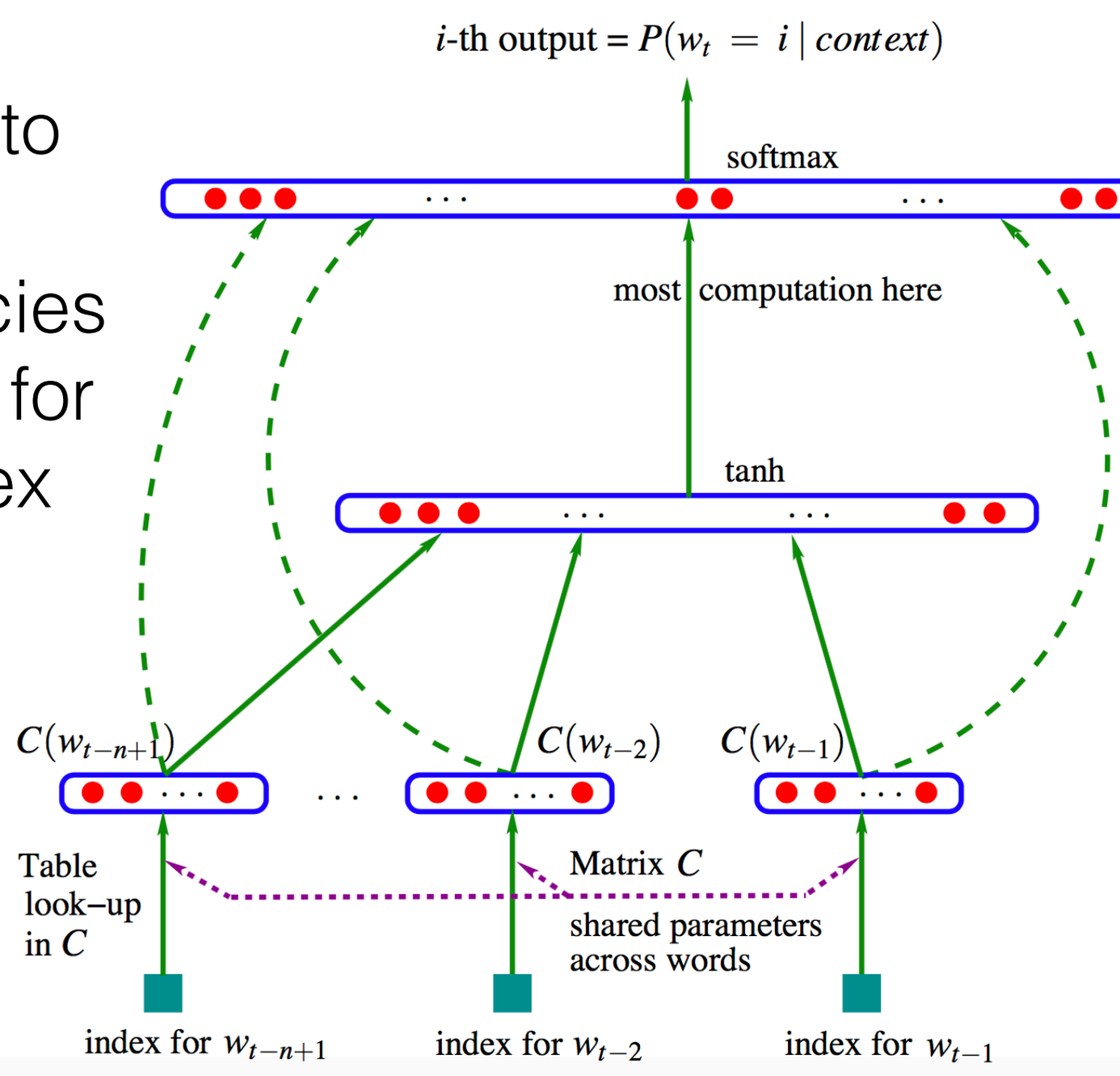
Limitations of the Bengio et al 2003 model

- This model still makes the Markov assumption, which ignores long-distance dependencies:
  - The **people** you saw at the grocery store last night **are** my friends.
  - I went to **Paris** but I didn't get a chance to see the rest of **France**.

Limitations of the Bengio et al 2003 model: Lack of temporal invariance

Loss: $-log\hat{P}(w)$

W needs to encode dependencies separately for each index



$U^{h\times|V|}$

$W^{(n-1)d\times h}$

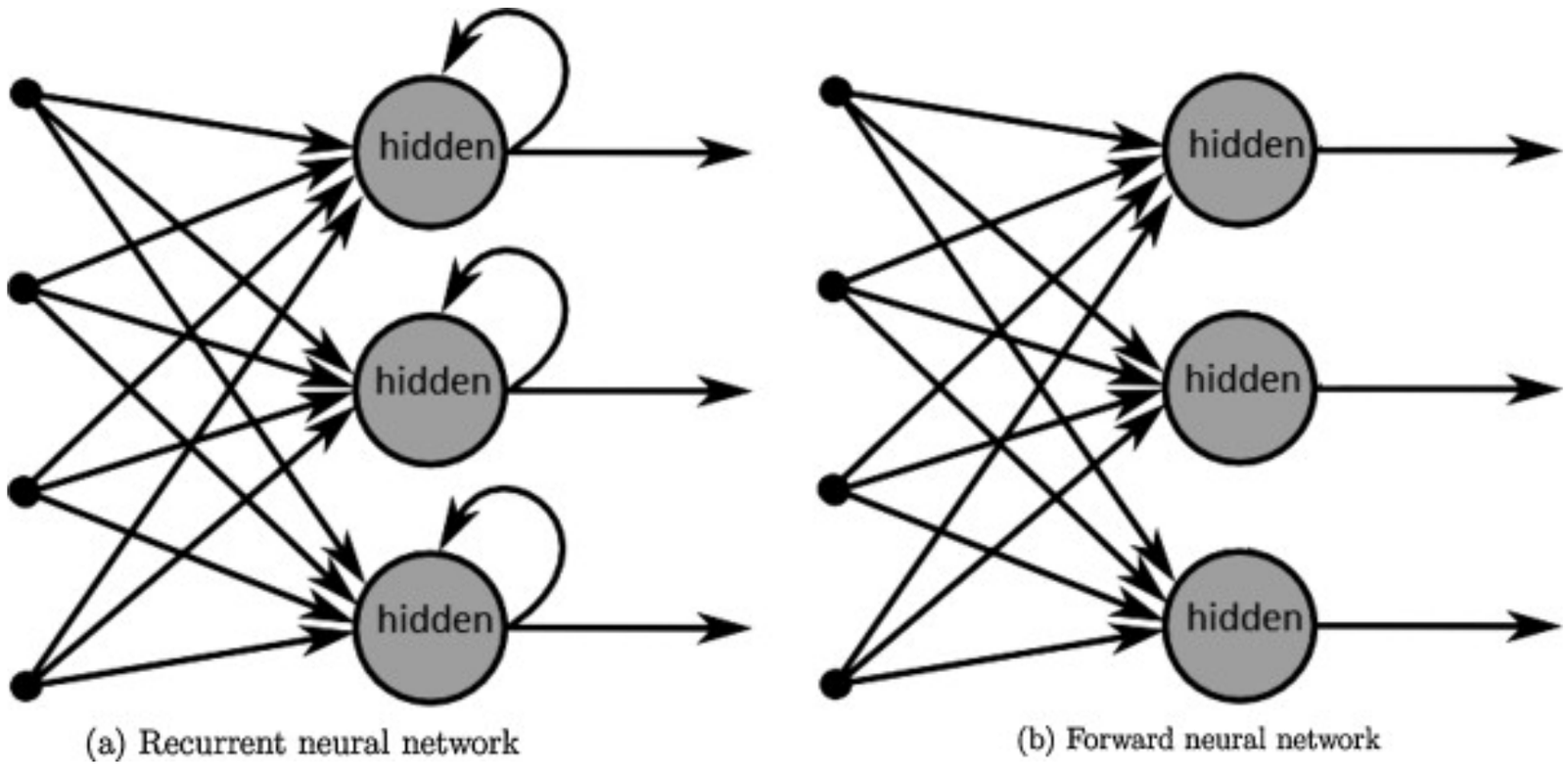
d-dimensional embeddings:

$C_1, \ldots, C_{|V|}$

(Bengio et al, 2003)

Simple recurrent network

$$\vec{h}_t = U\vec{i}_t + W\vec{h}_{t-1} \qquad \vec{h}_t = U\vec{i}_t$$

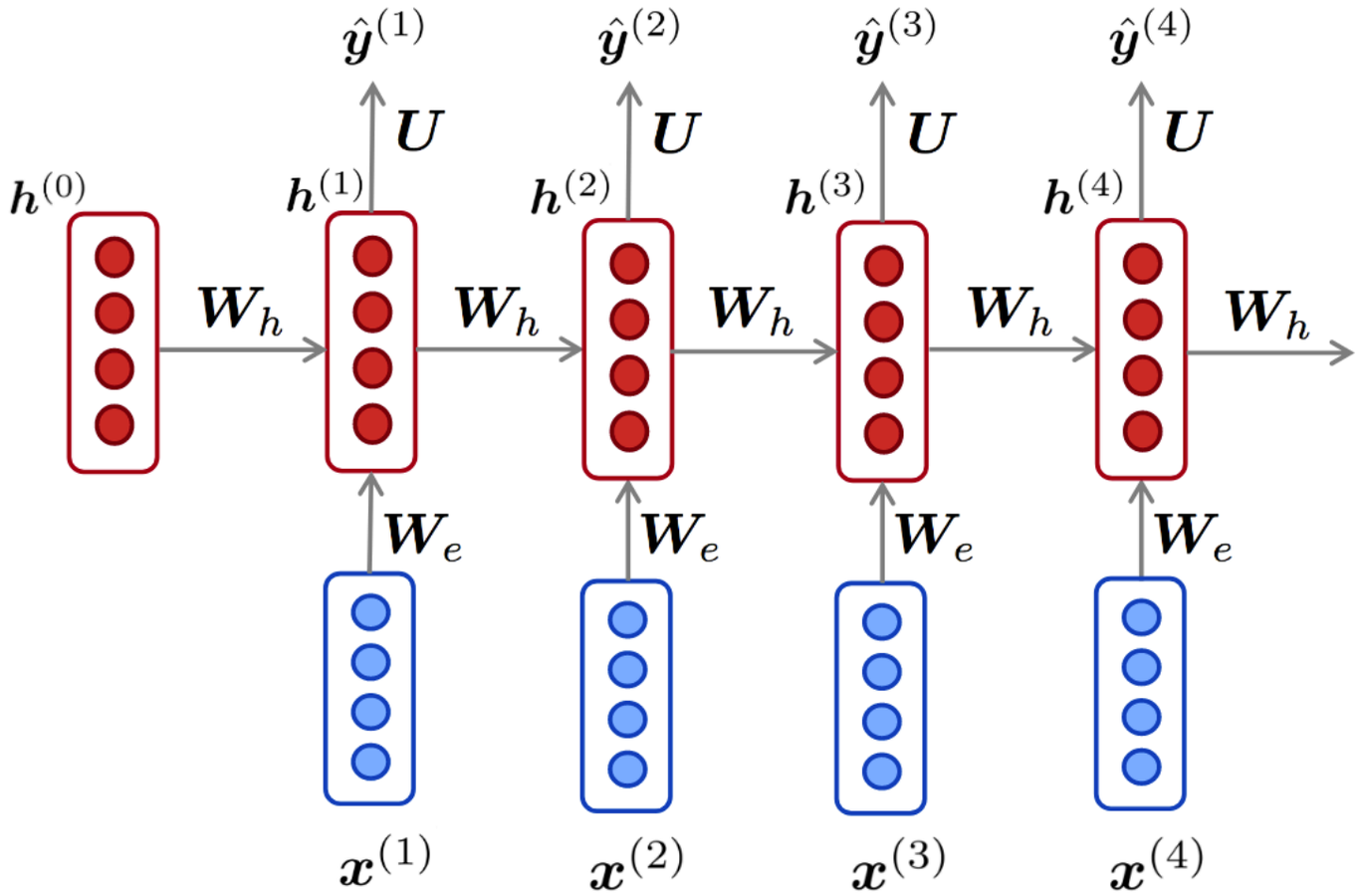


(a) Recurrent neural network (b) Forward neural network

(Recurrent model) (Feedforward model)

(figure from Mulder et al., 2015)

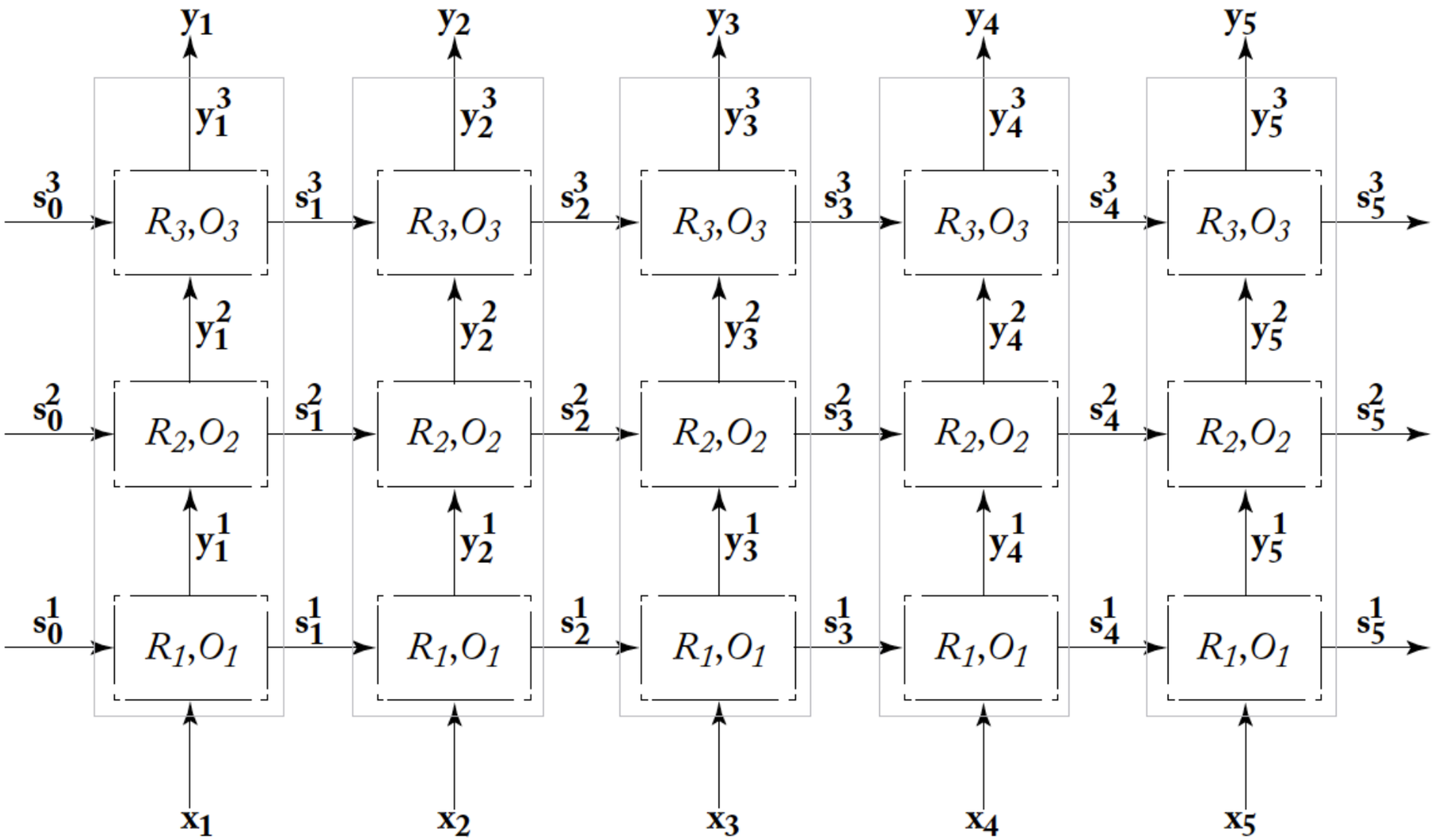
Unrolling an RNN



$$\frac{\partial L}{\partial W_h} = \sum_{t=1}^{T} \frac{\partial L_t}{\partial W_h}$$

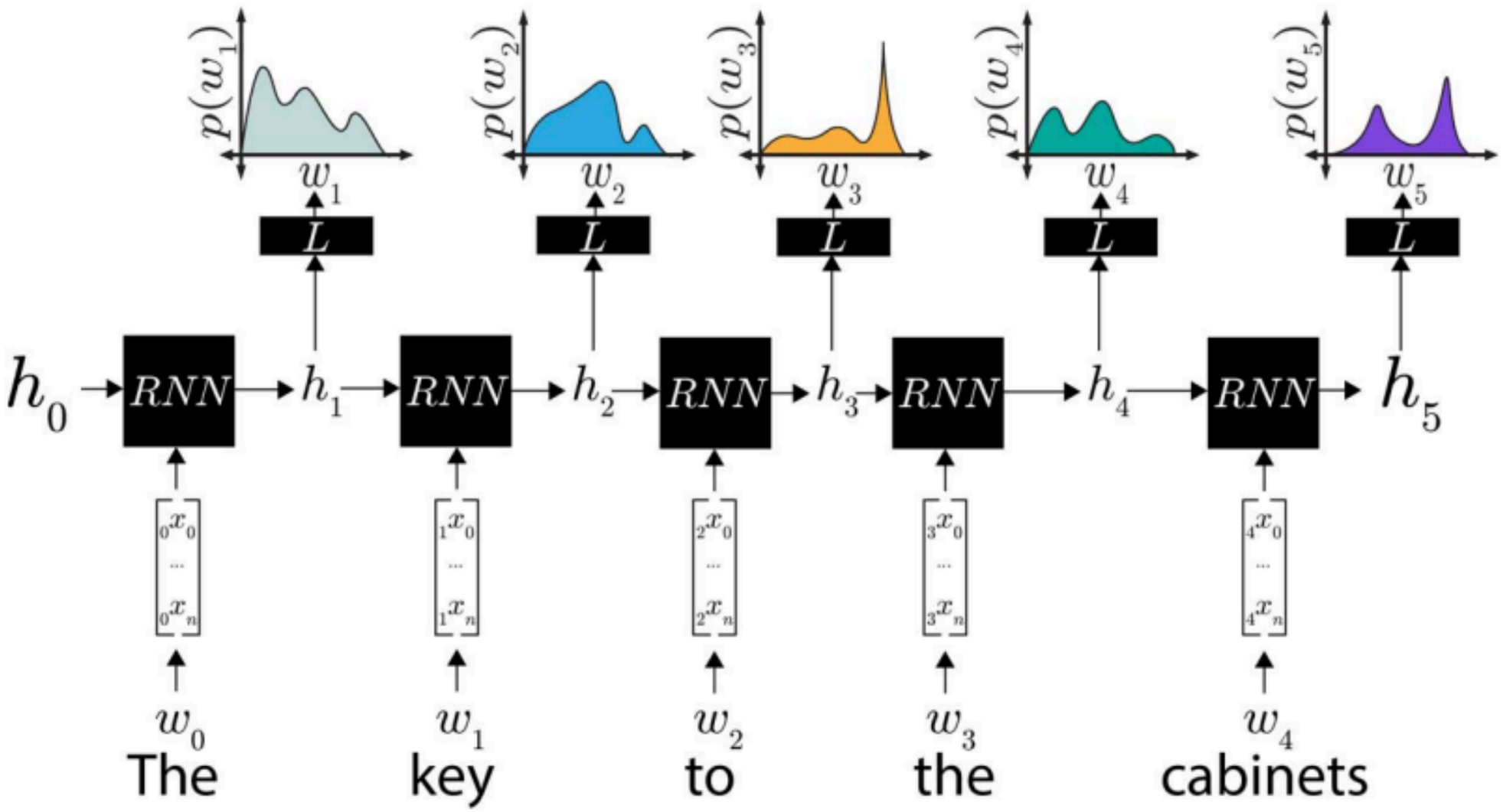
(Figure credit: Richard Socher)

Stacked RNNs



(From Goldberg 2017)

RNNs as language models



A slide you do not need to understand: LSTM ("long short-term memory")

$$c_t = f_t c_{t-1} + i_t g_t$$

$$z_t = \textbf{concat}(D_{t-1}, x_t)$$

$$i_t = \sigma(W_i z_t + b_i)$$

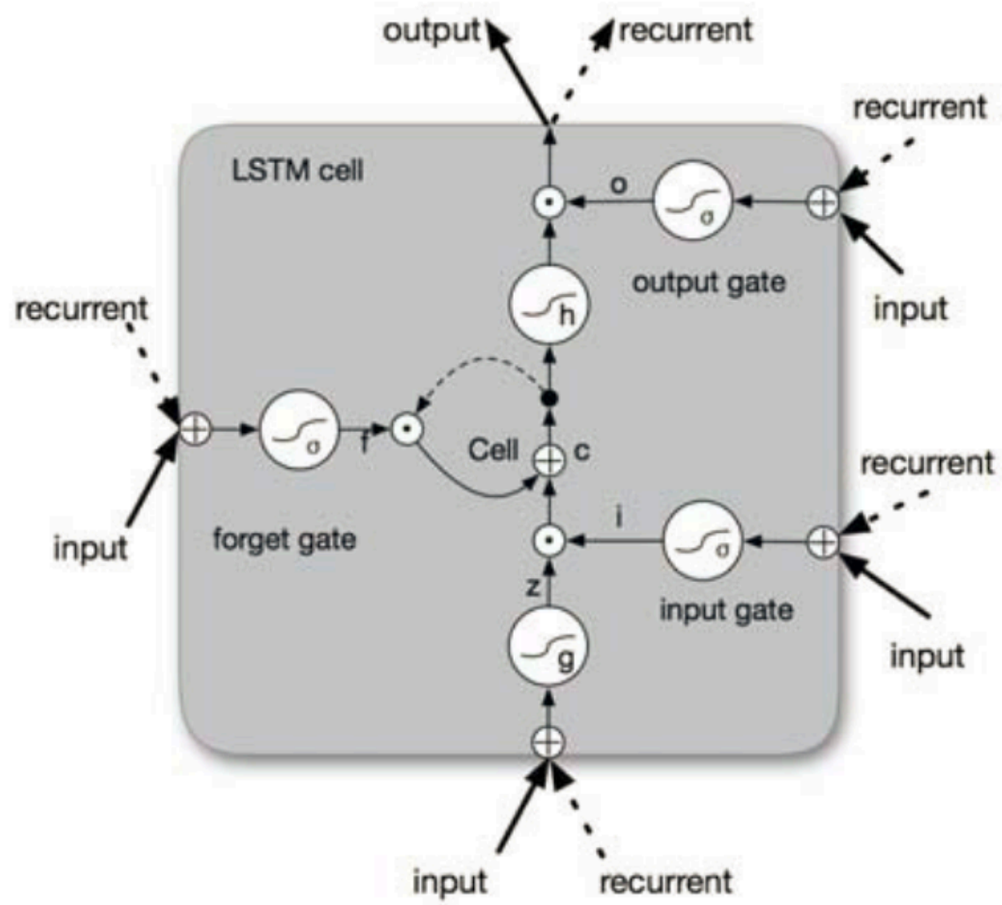
$$f_t = \sigma(W_f z_t + b_f)$$

$$g_t = \tanh(W_g z_t + b_g)$$

$$D_t = o_t \tanh(c_t)$$

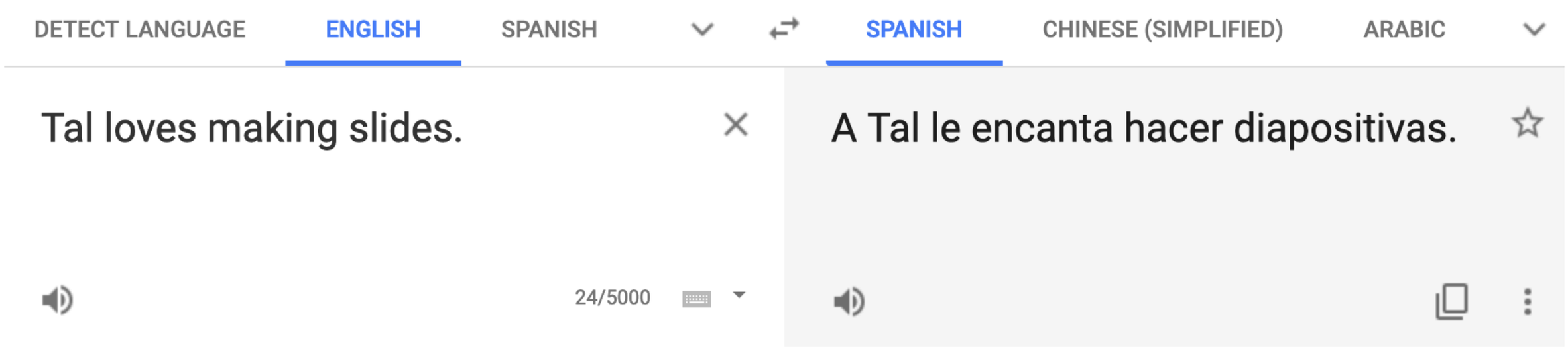
$$o_t = \sigma(W_o z_t + b_o)$$

(Hochreiter & Schmidhuber 1997; figure from Ma & Hovy 2016)



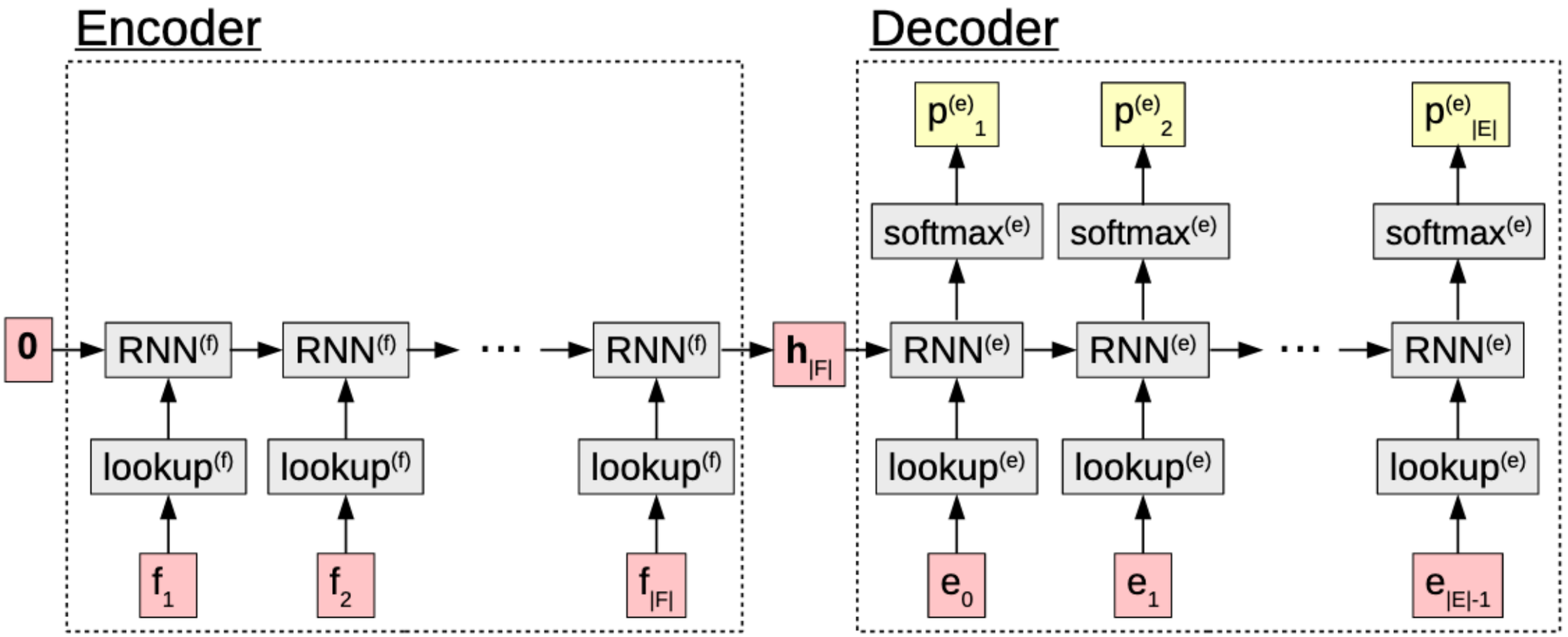
Conditional language modeling

Conditioned language modeling (e.g. for machine translation)



- Spanish unconditioned language model: $P(w_n \mid w_1 \ldots, w_{n-1})$
- Conditioned language model (Spanish given English): $P(w_n \mid x, w_1 \ldots, w_{n-1})$ (where x is the English sentence)
- Goal of machine translation: find $\arg\max_{y \in Y} P(y \mid x)$

Encoder/decoder RNN



(Neubig 2017)

$$\boldsymbol{m}_t^{(f)} = M_{\cdot,f_t}^{(f)}$$

$$\boldsymbol{h}_t^{(f)} = \begin{cases} \text{RNN}^{(f)}(\boldsymbol{m}_t^{(f)}, \boldsymbol{h}_{t-1}^{(f)}) & t \geq 1, \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

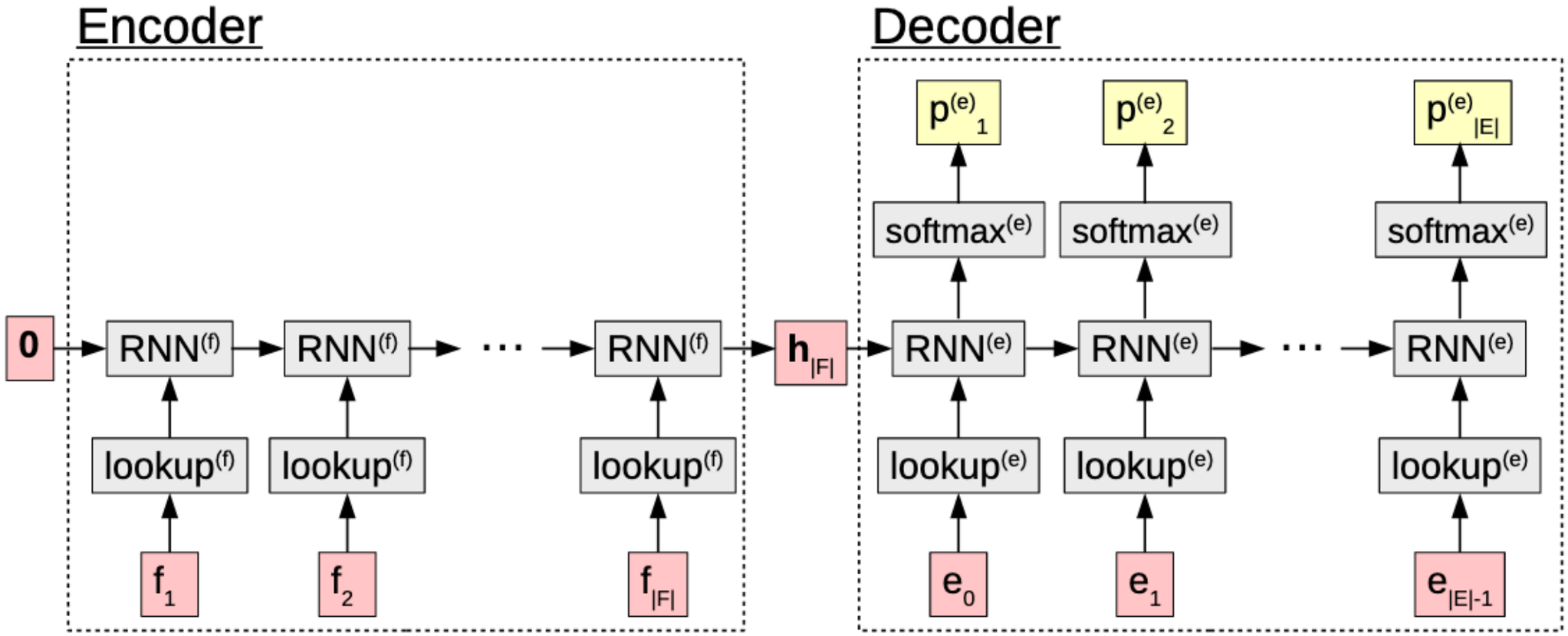
$$\boldsymbol{m}_t^{(e)} = M_{\cdot,e_{t-1}}^{(e)}$$

$$\boldsymbol{h}_t^{(e)} = \begin{cases} \text{RNN}^{(e)}(\boldsymbol{m}_t^{(e)}, \boldsymbol{h}_{t-1}^{(e)}) & t \geq 1, \\ \boldsymbol{h}_{|F|}^{(f)} & \text{otherwise.} \end{cases}$$

$$\boldsymbol{p}_t^{(e)} = \text{softmax}(W_{hs}\boldsymbol{h}_t^{(e)} + b_s)$$

In training the decoder normally uses "teacher forcing": $\boldsymbol{e}_i$ is the word that occurred in the corpus, not the recently predicted $\boldsymbol{p}_{i-1}^{(e)}$

Decoding strategies: random sampling



Sample from $P(y \mid x)$:

Sample a word from $p_t^{(e)}$ until we sample </s>

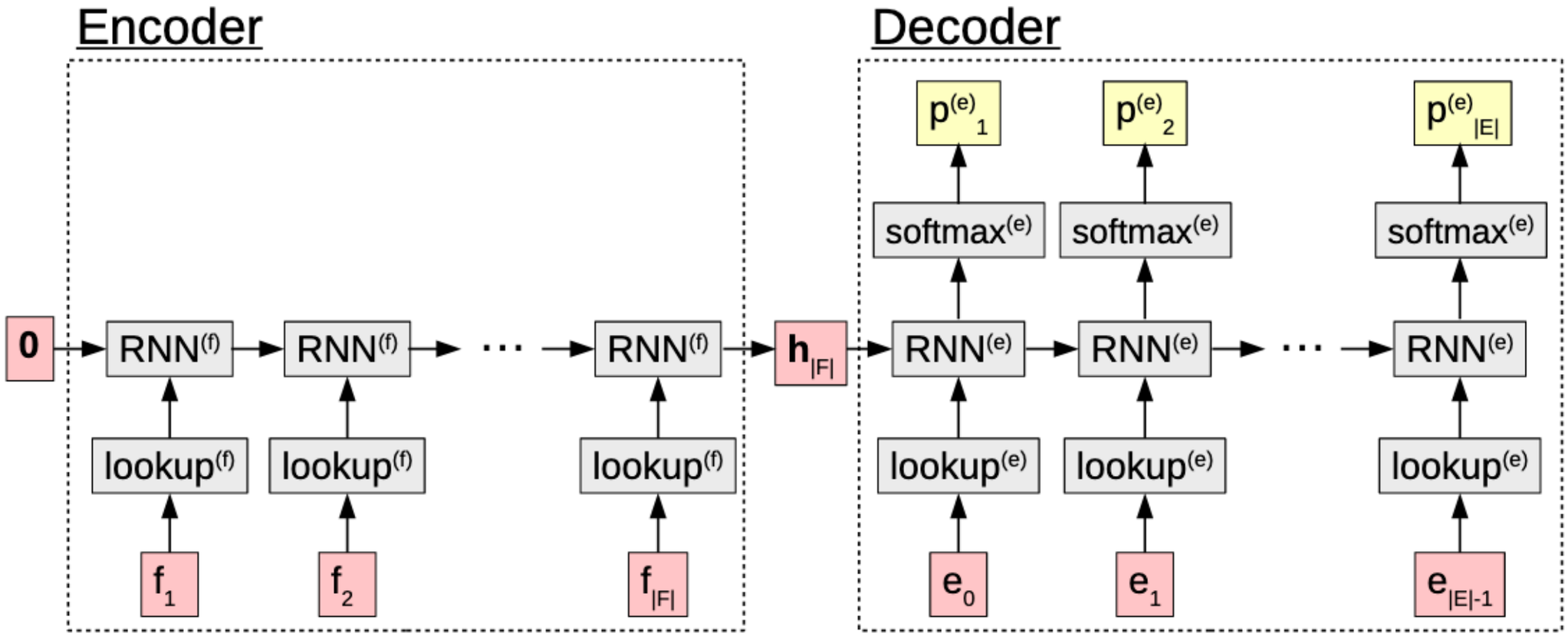
Decoding strategies: random sampling

- The output is stochastic (different every time we repeat the process)

- Variability in generation outputs can be good (for example in dialog)

- This method will often prefer to generate a sequence of high-frequency words over a single infrequent one

Decoding strategies: greedy decoding

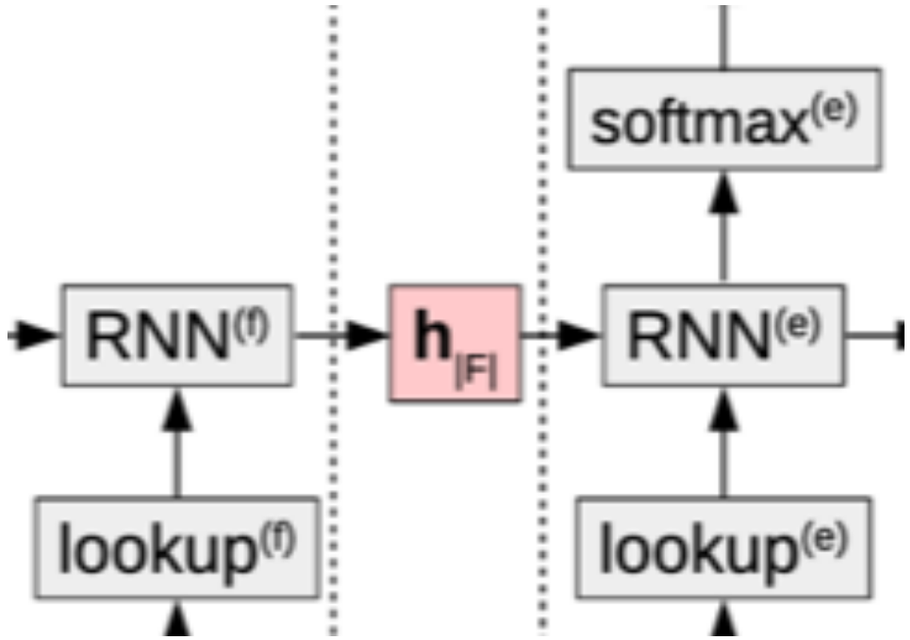


The output at time step t is $\arg\max_{1 \le i \le |V|} p_{t,i}^{(e)}$

(still until we sample </s>)

Attention and transformers

Attention: motivation



- A single hidden state needs to represent all of the sentence so far - that's a heavy burden!
- Why limit ourselves to a fixed-size vector bottleneck when we can peek at all of the RNN's previous states?

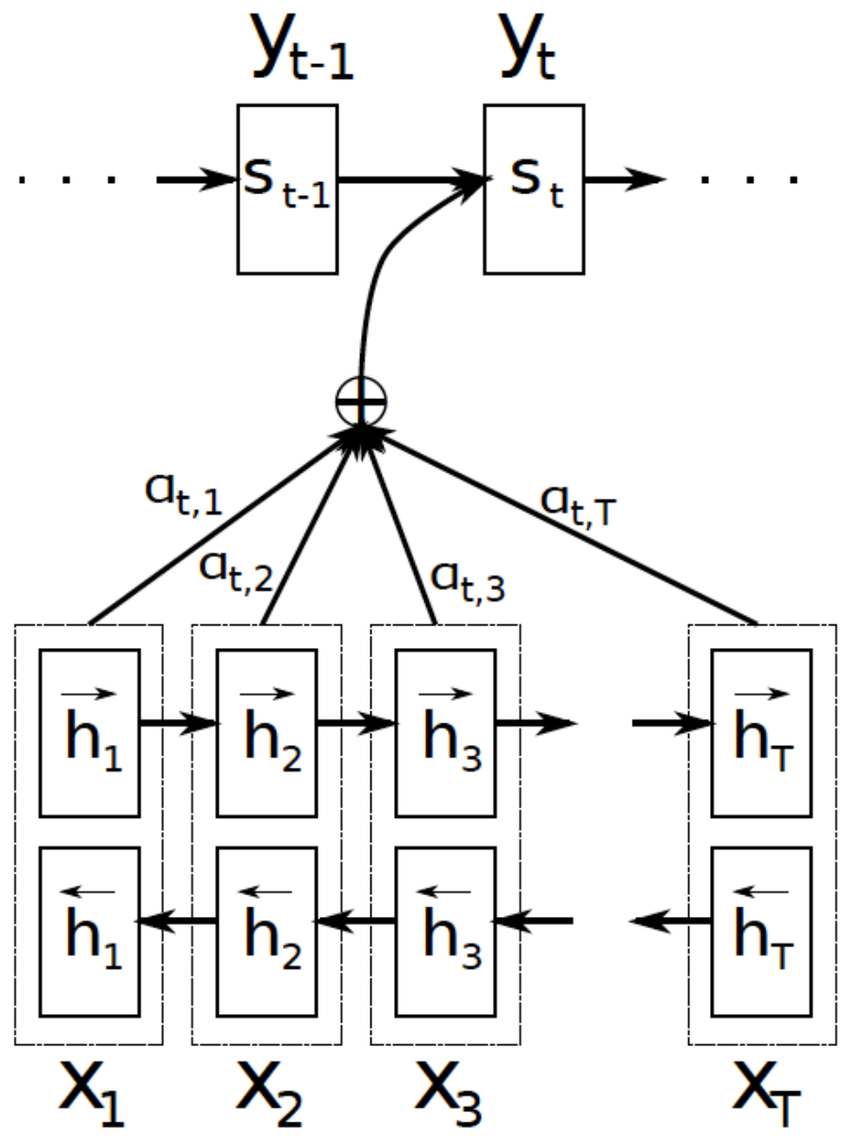
How much should we care about each of the encoder's hidden states?

Attention function

$$e_{ij} = a(s_{i-1}, h_j)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_j \exp(e_{ij})}$$

Softmax



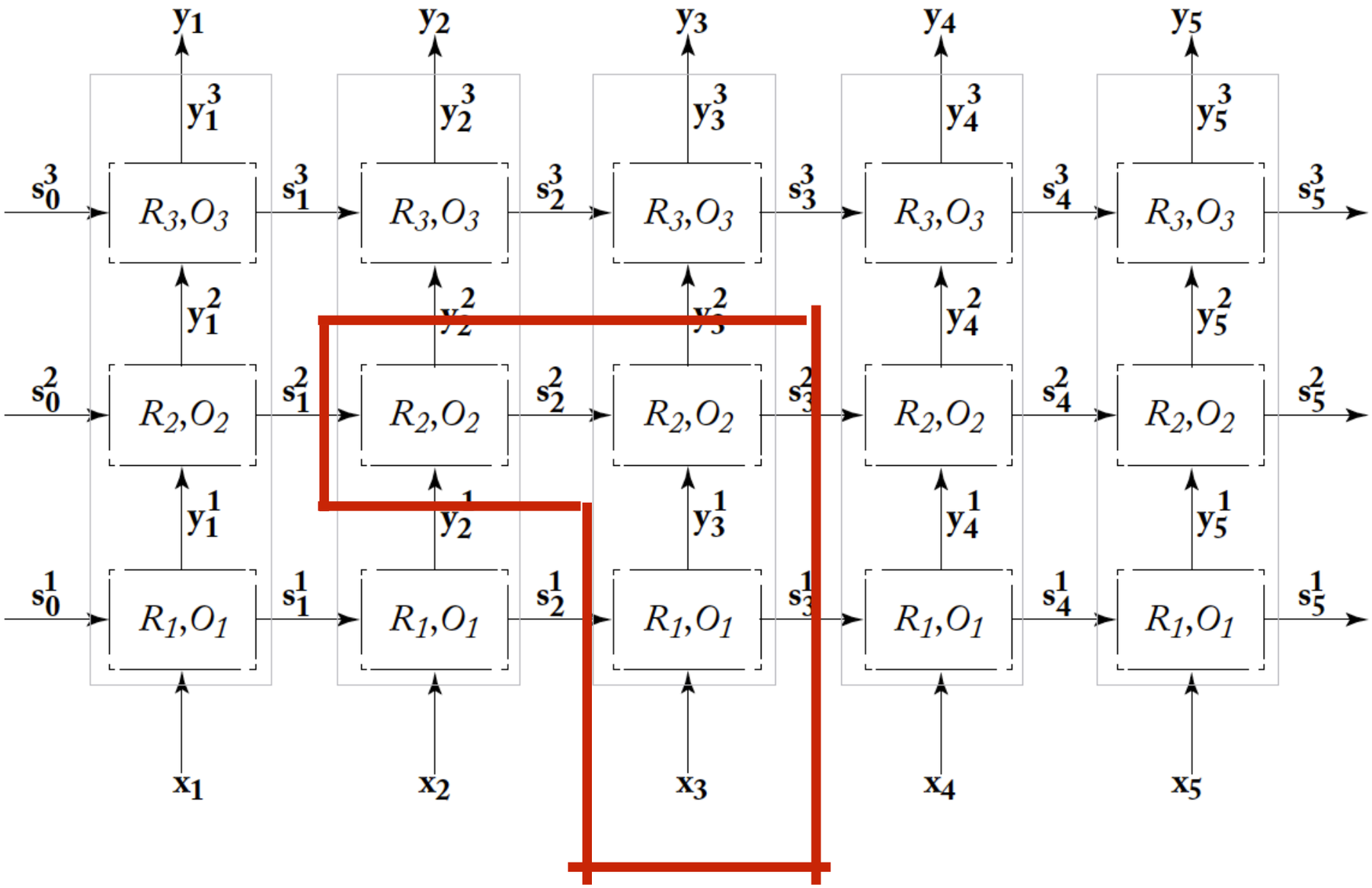
(Bahdanau et al., 2015)

Attention functions

- Let's refer to the decoder's state as the "query" and to the encoder states as the "keys"
- MLP: $a(q,k) = u^T \tanh(W[q;k])$ (Bahdanau et al)
- Dot product: $a(q,k) = q^T k$
  - Simpler: no learned parameters (but query and key need to be the same size)
- Scaled dot product (used in transformers) keeps dot product from scaling with the size of the vectors:

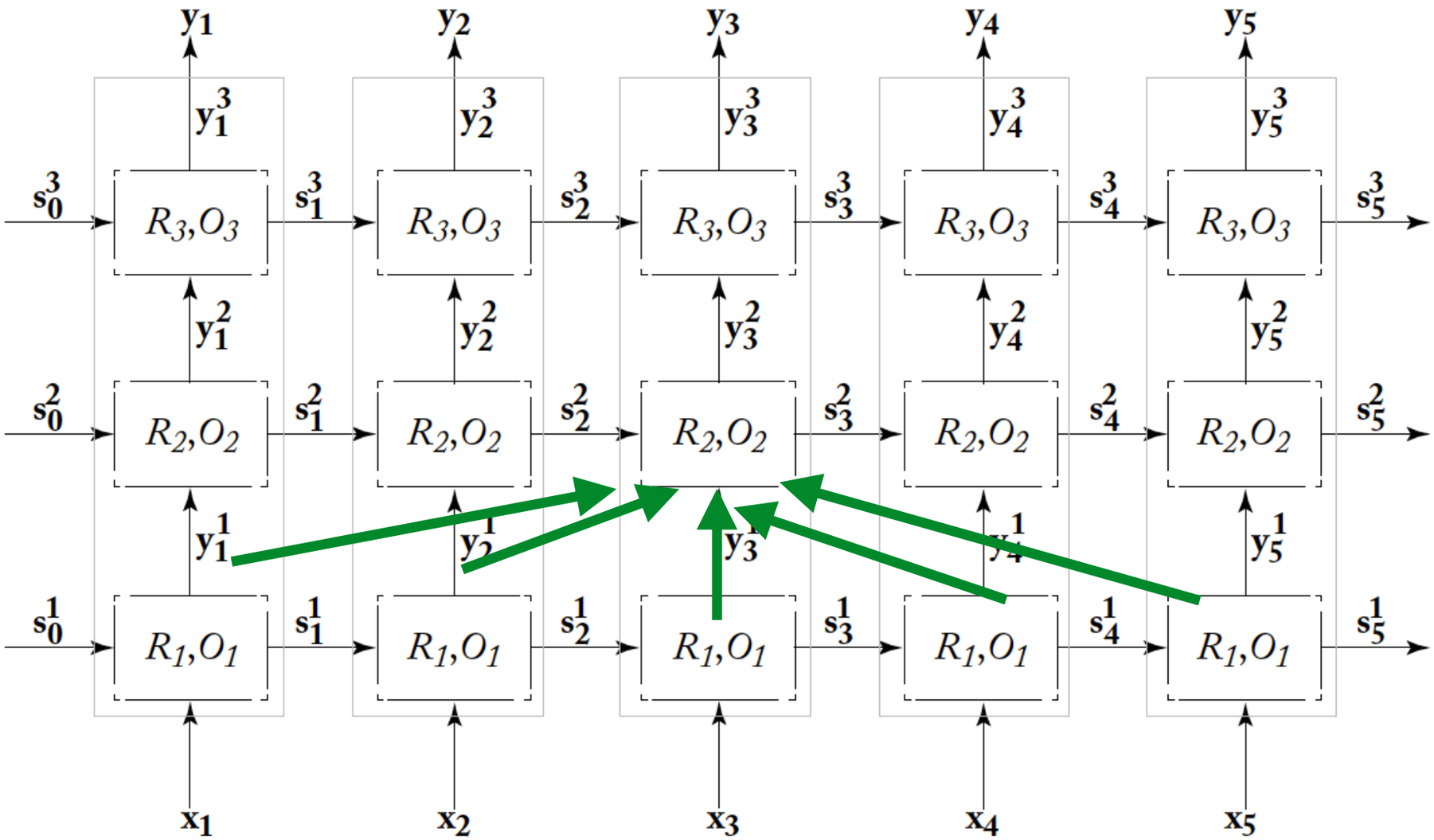
$$a(q,k) = \frac{q^T k}{\sqrt{|q|}}$$

Motivating self-attention: a stacked RNN as the encoder



(From Goldberg 2017)

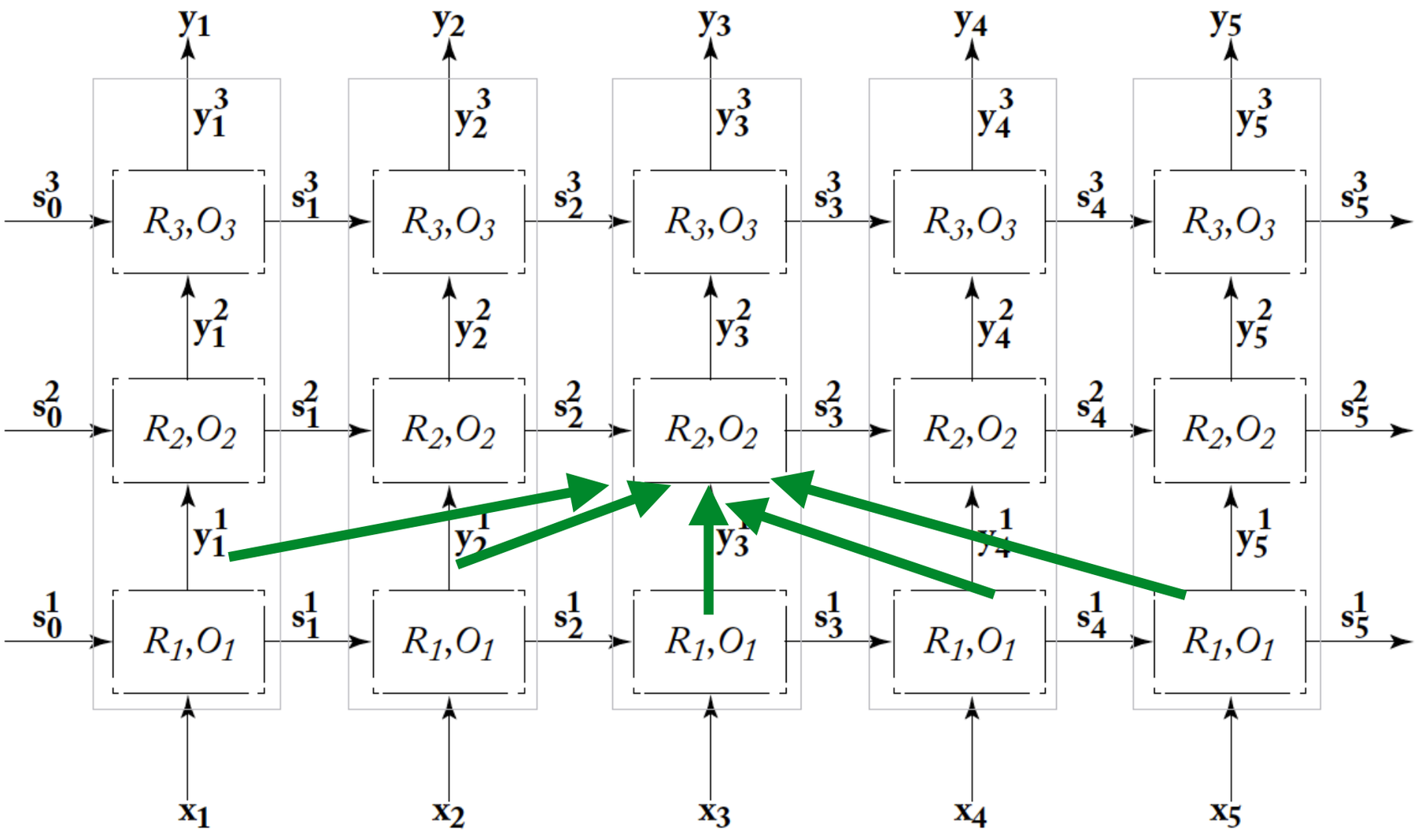
Motivating self-attention: a stacked RNN as the encoder



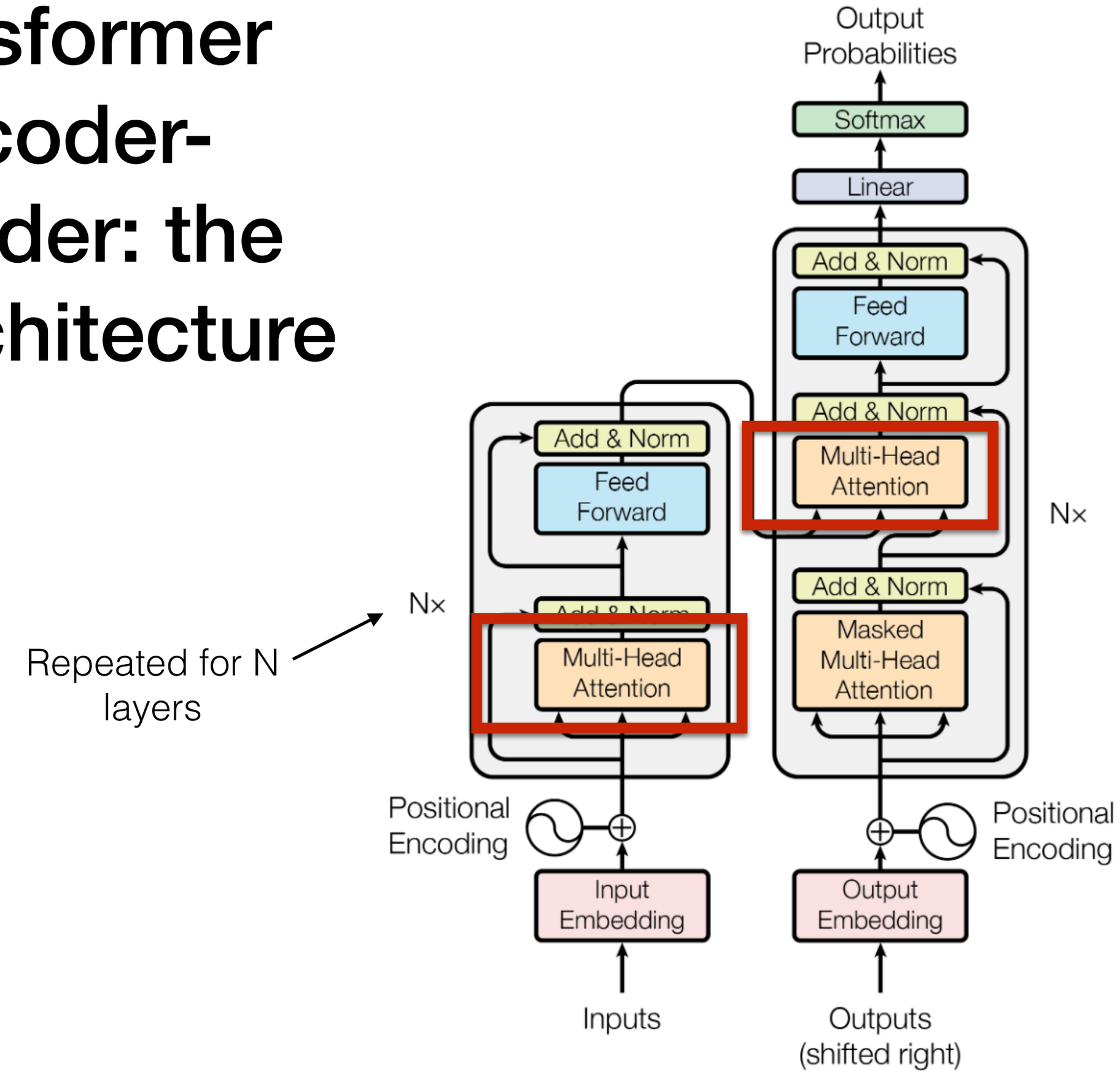
(From Goldberg 2017)

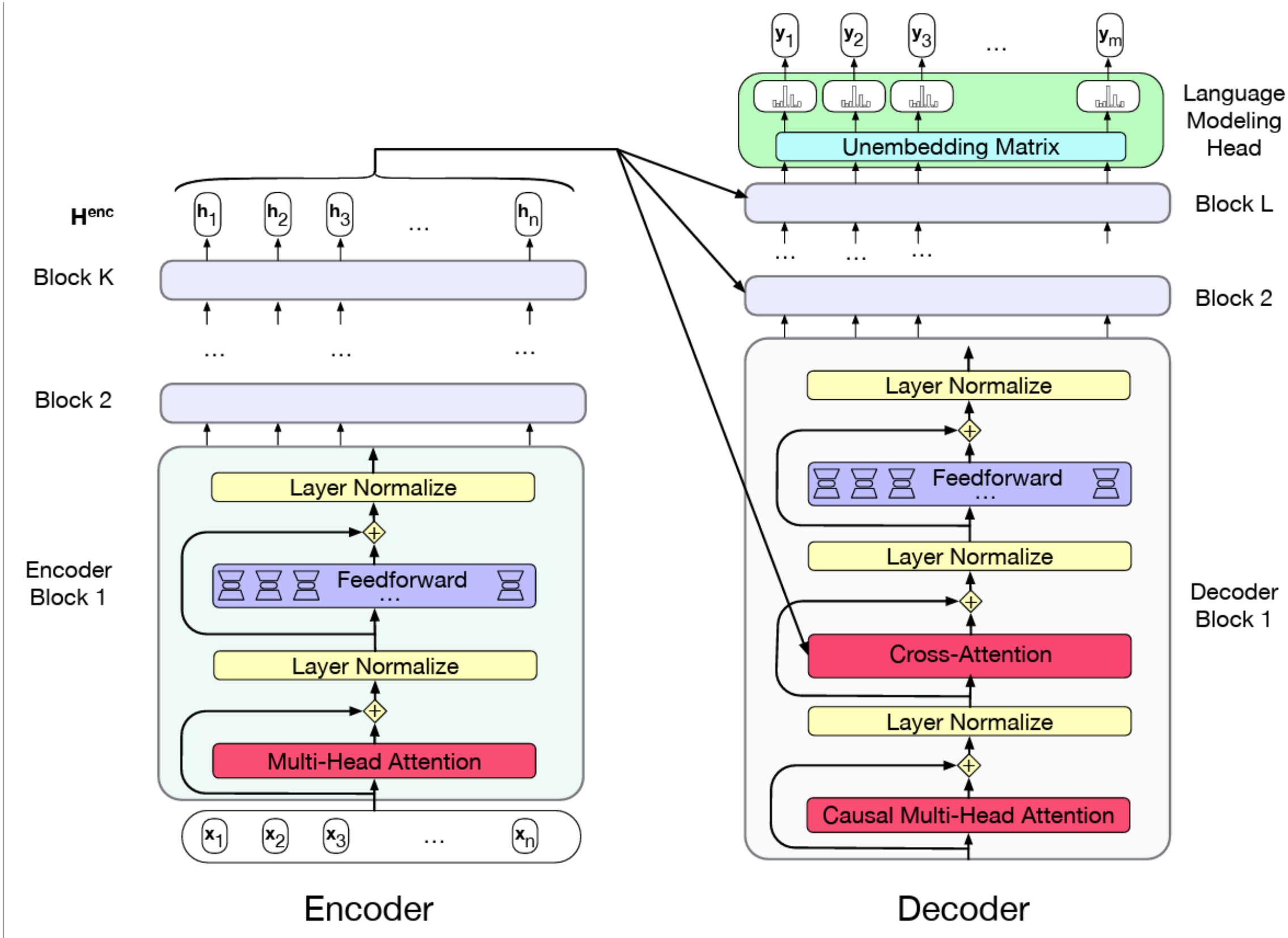
A stacked RNN with self-attention as the encoder

The decoder typically attends to the output of the top layer of the encoder



Transformer encoder-decoder: the full architecture




y_1 y_2 y_3 … y_m
Unembedding Matrix
Language Modeling Head
H^{enc}
h_1 h_2 h_3 … h_n
Block K
Block L
Block 2
Layer Normalize
Feedforward
Cross-Attention
Causal Multi-Head Attention
Encoder Block 1
Decoder Block 1
x_1 x_2 x_3 … x_n
Encoder
Decoder

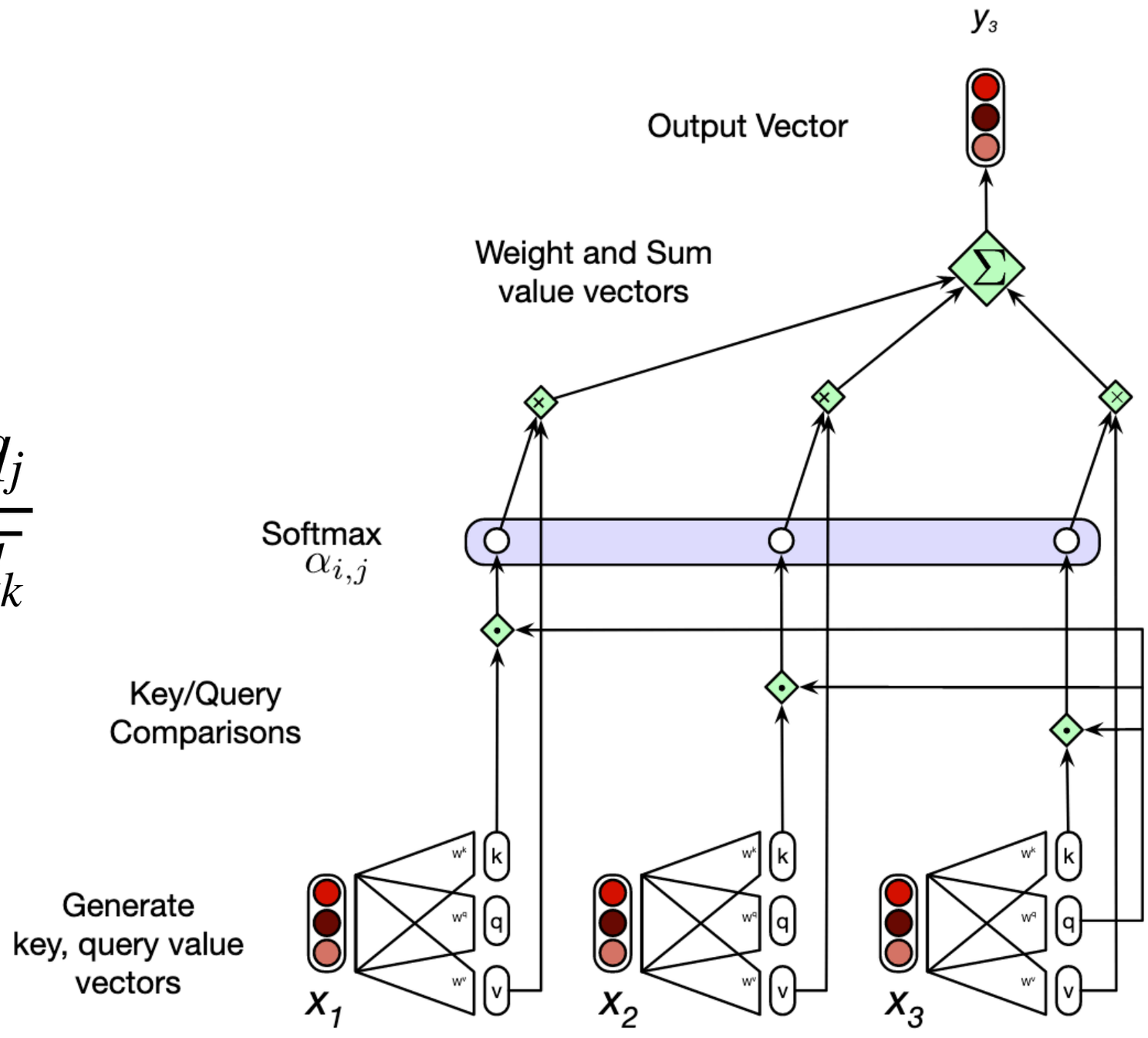
Transformer self-attention

$$q_i = W^Q x_i$$

$$k_i = W^K x_i$$

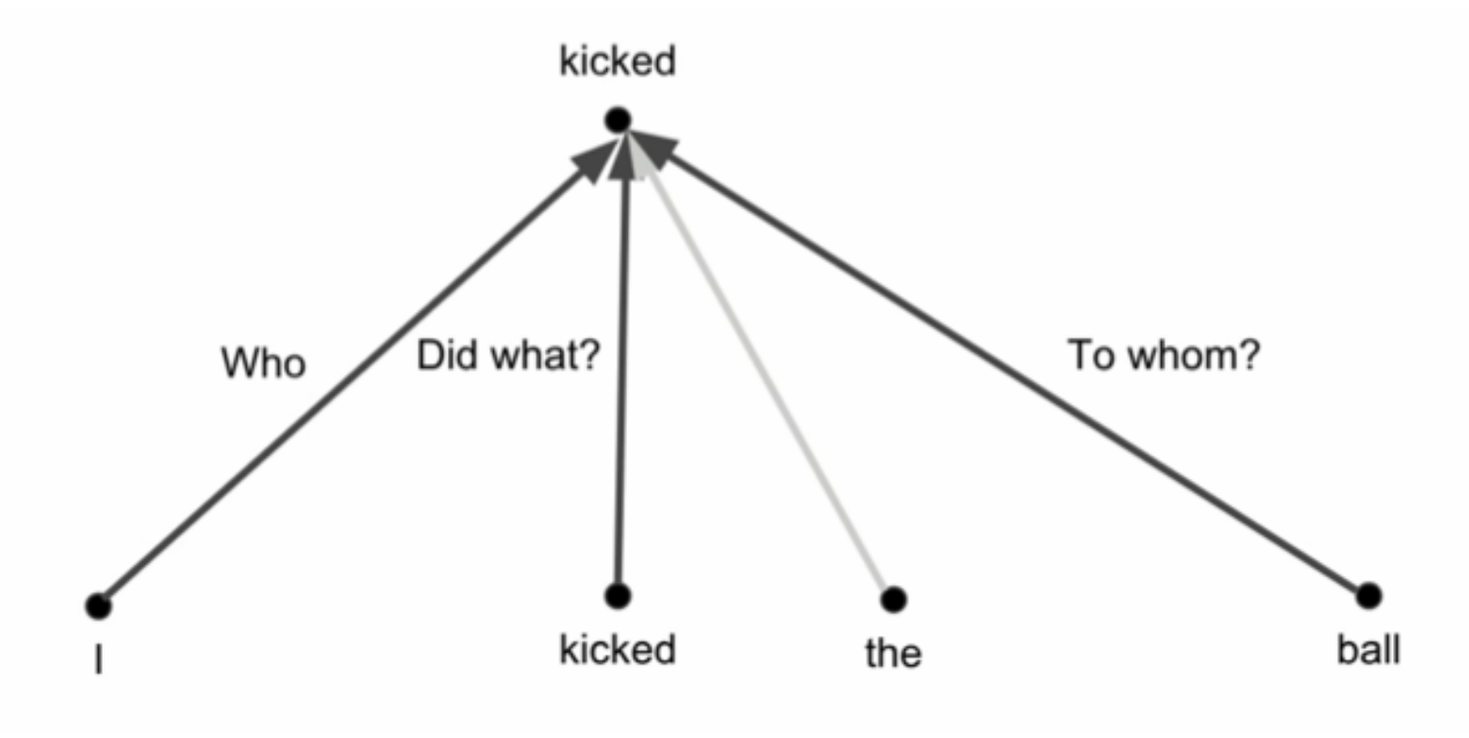
$$v_i = W^V x_i$$

$$\text{score}(x_i, x_j) = \frac{k_i^T q_j}{\sqrt{d_k}}$$

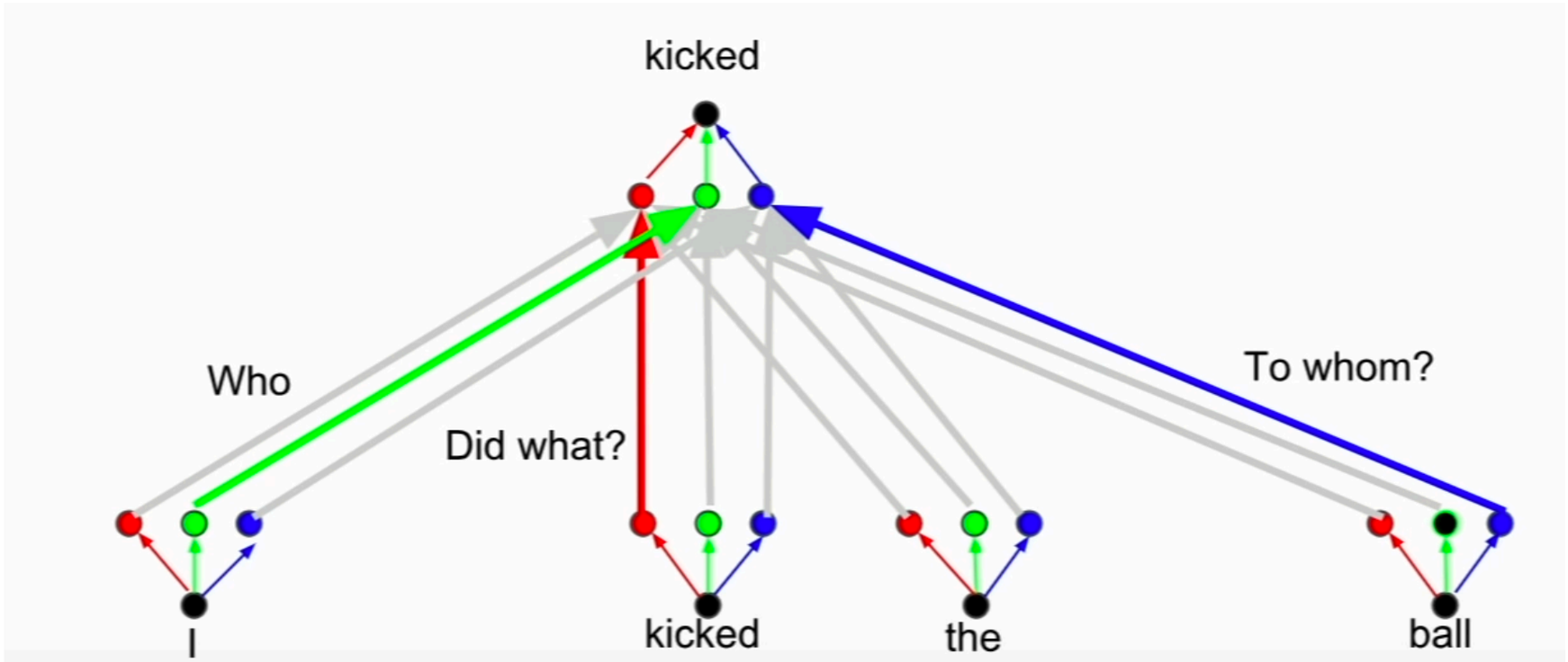


Multi-head attention: motivation

- Attention treats all linguistic relations alike: here, the agent (who did the action), the verb, and the patient (who received the action) all need to be combined with the same W^Q, W^V and W^K



Let’s let the model learn different types of attention (“attention heads”)!



Let's let the model learn different types of attention ("attention heads")!

concatenation

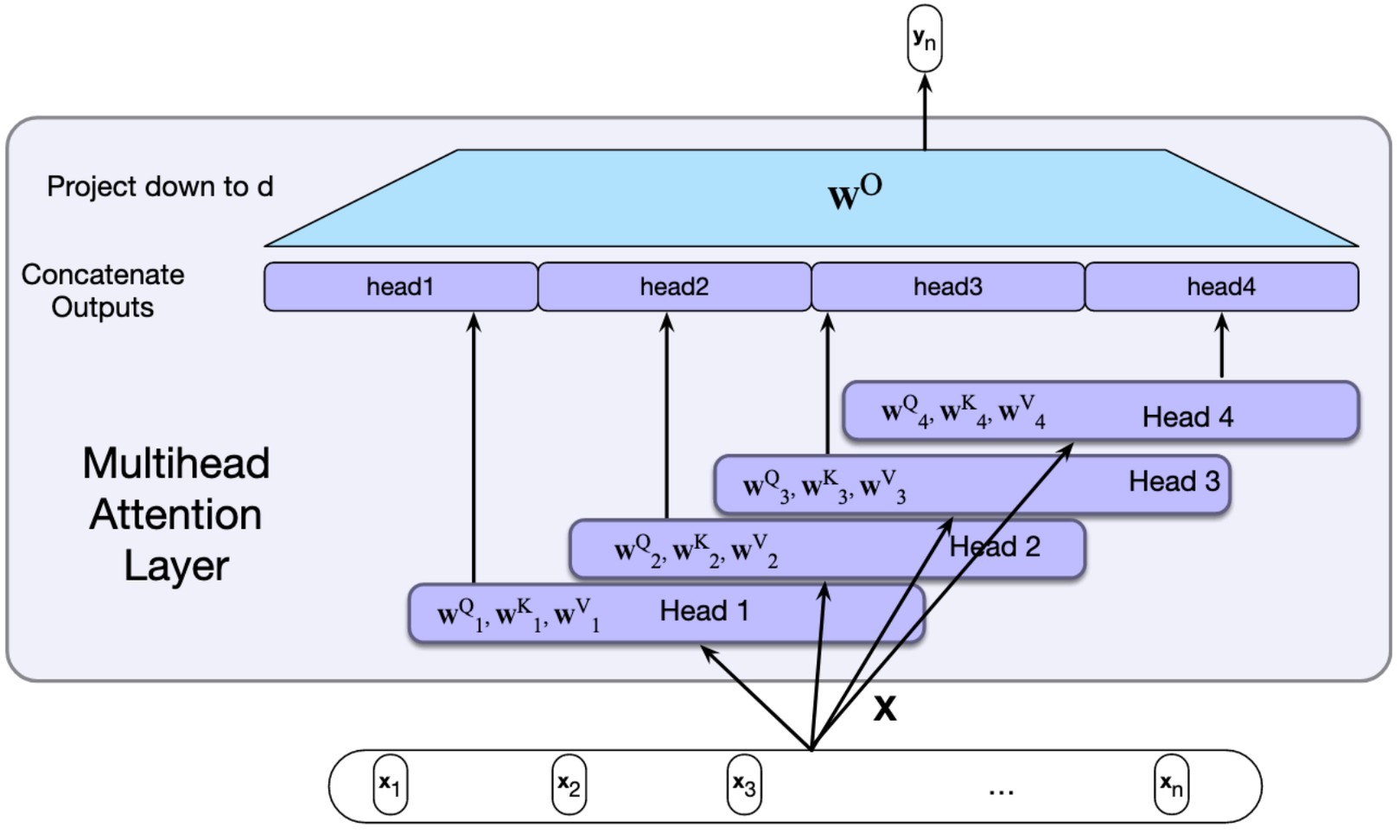
$$Y = [h_1; h_2; \ldots; h_k]W^O$$

$$h_i = \text{softmax}(Q_i, K_i, V_i)$$

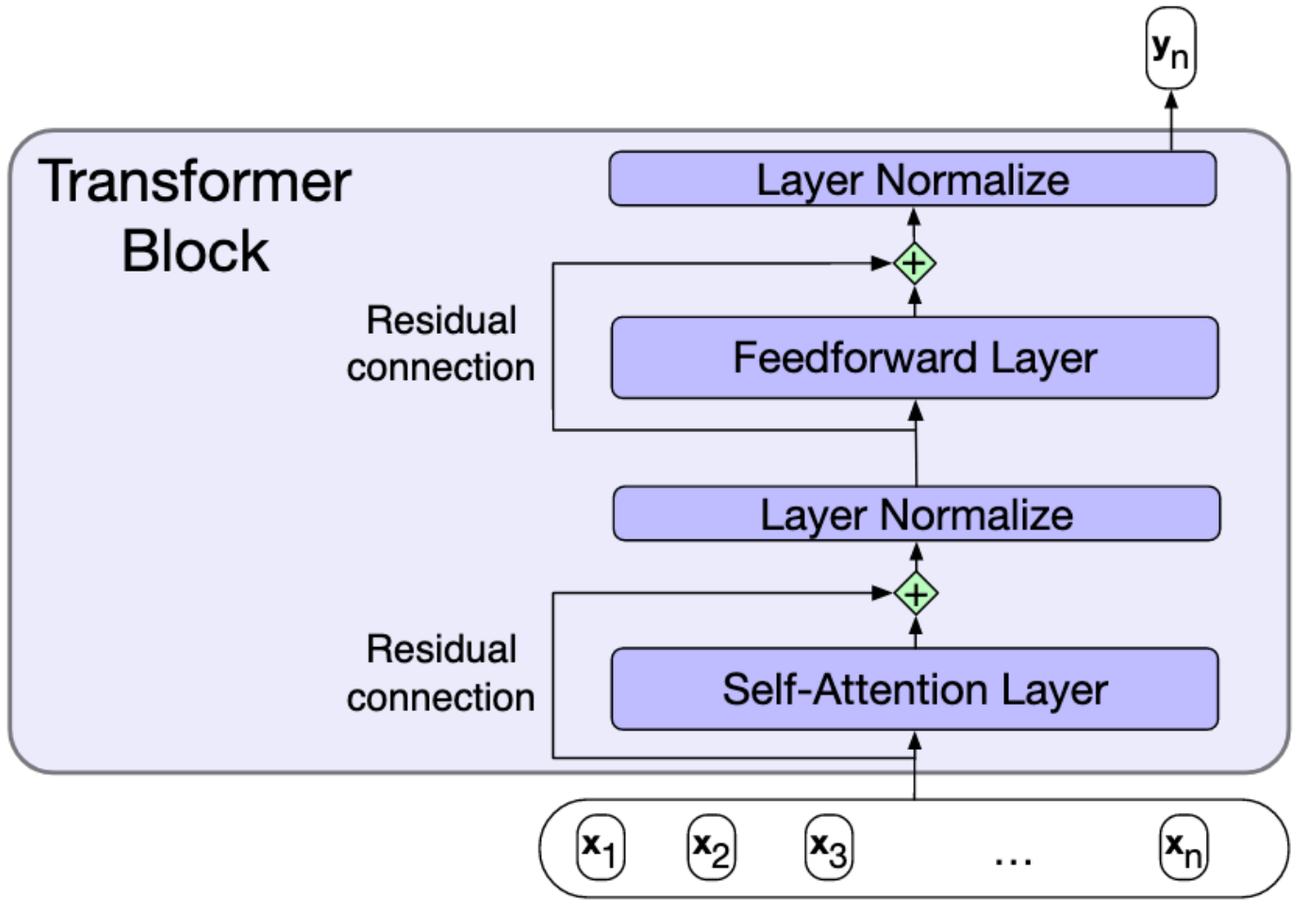
$$Q_i = XW^{Q_i}$$

$$K_i = XW^{K_i}$$

$$V_i = XW^{V_i}$$



Attention is not quite all you need

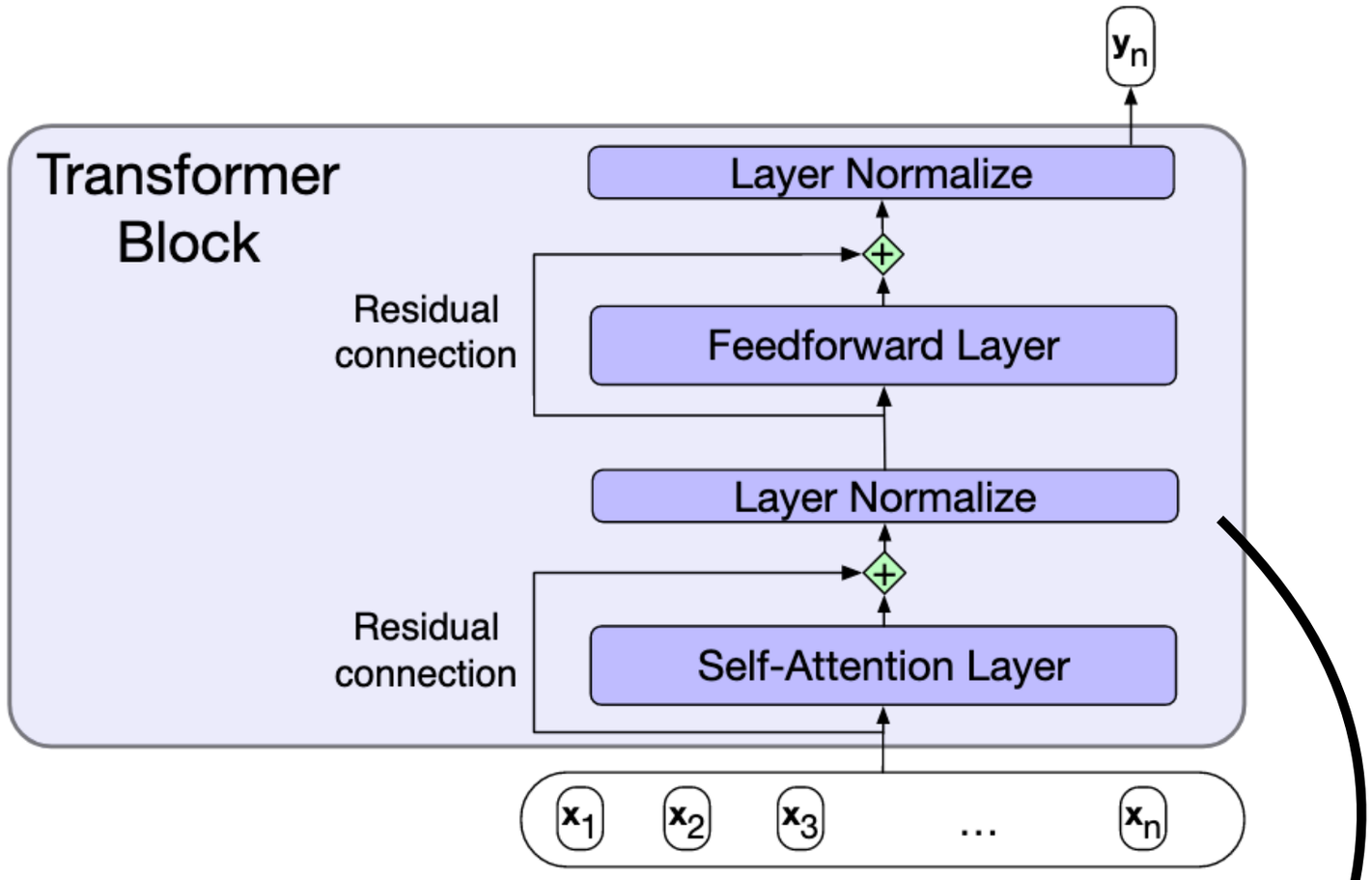


Layer norm: $\hat{x} = \frac{x - \mu}{\sigma}$

Mean of all units in layer

Standard deviation of all units in layer

Attention is not quite all you need



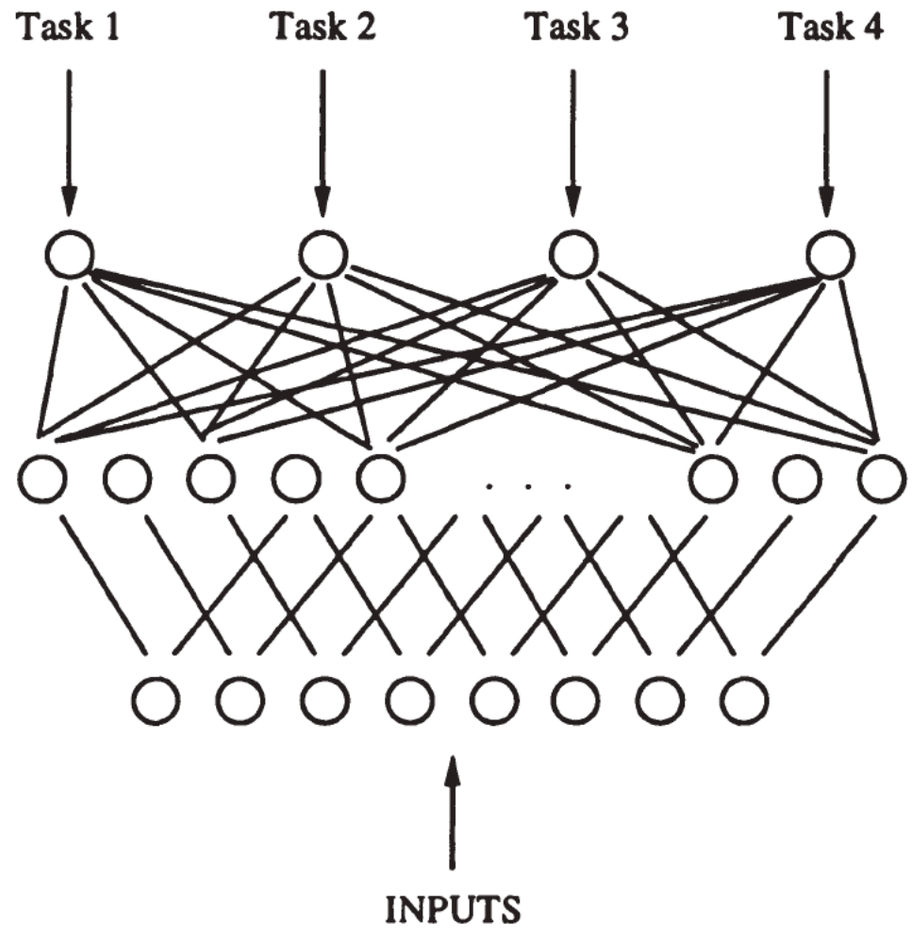
LayerNorm(x+SelfAttention(x))

Residual connection

Synergy between different machine learning tasks: representation sharing

- Transfer learning: using representations from a data-rich task to bootstrap represenations learning for a data-poor one

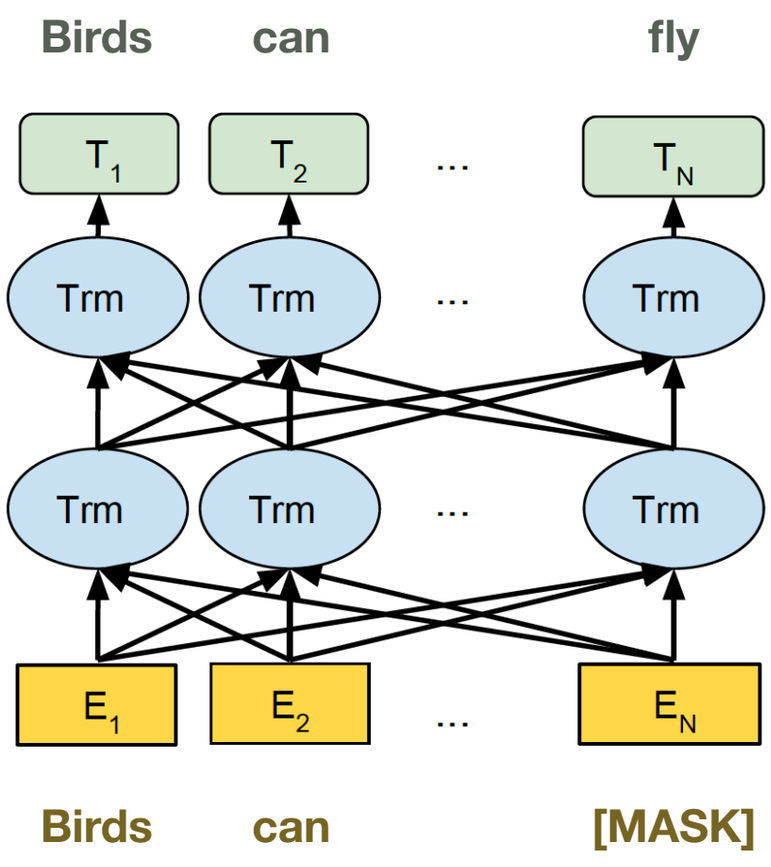
- Multi-task learning: jointly learn representations that are useful for multiple tasks at once



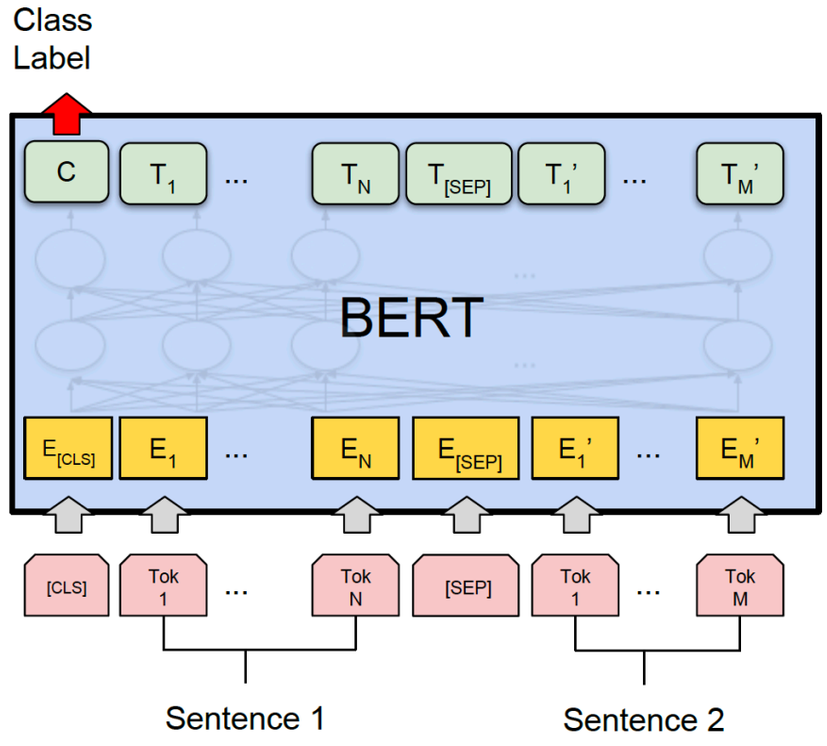
Multitask learning (Caruana, 1998)

Transfer learning from language modeling

1. Pre-train on word prediction (or other self-supervised objective) to obtained contextualized word representations



2. Fine tune on the training set for the supervised task



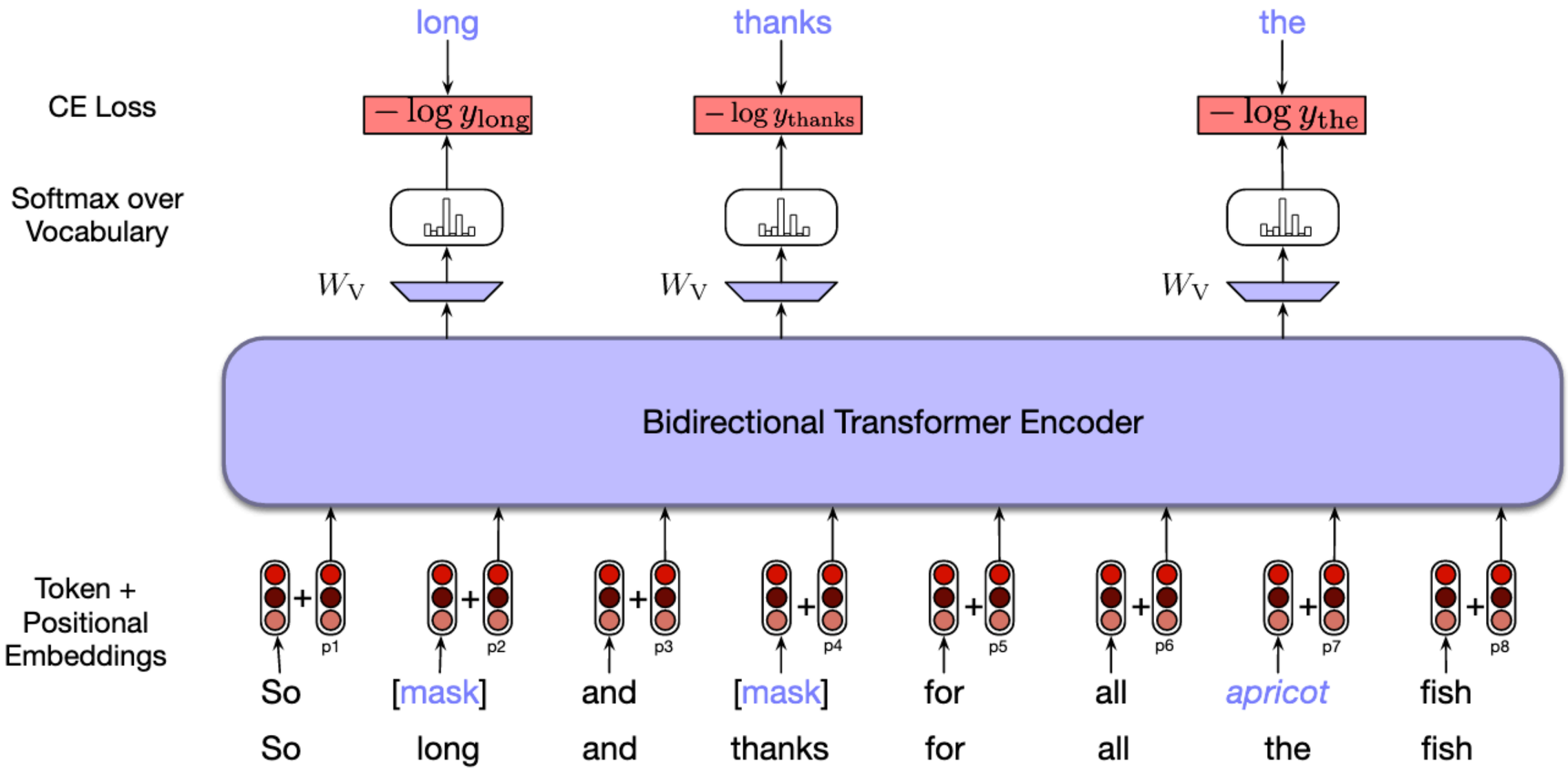
x_1 x_3 x_4 x_6 x_n

3. Evaluate on test set

Model	URL Score
ALBERT + DAAF + NAS	90.6
ERNIE	90.4

x_2 x_5 x_{n-1}

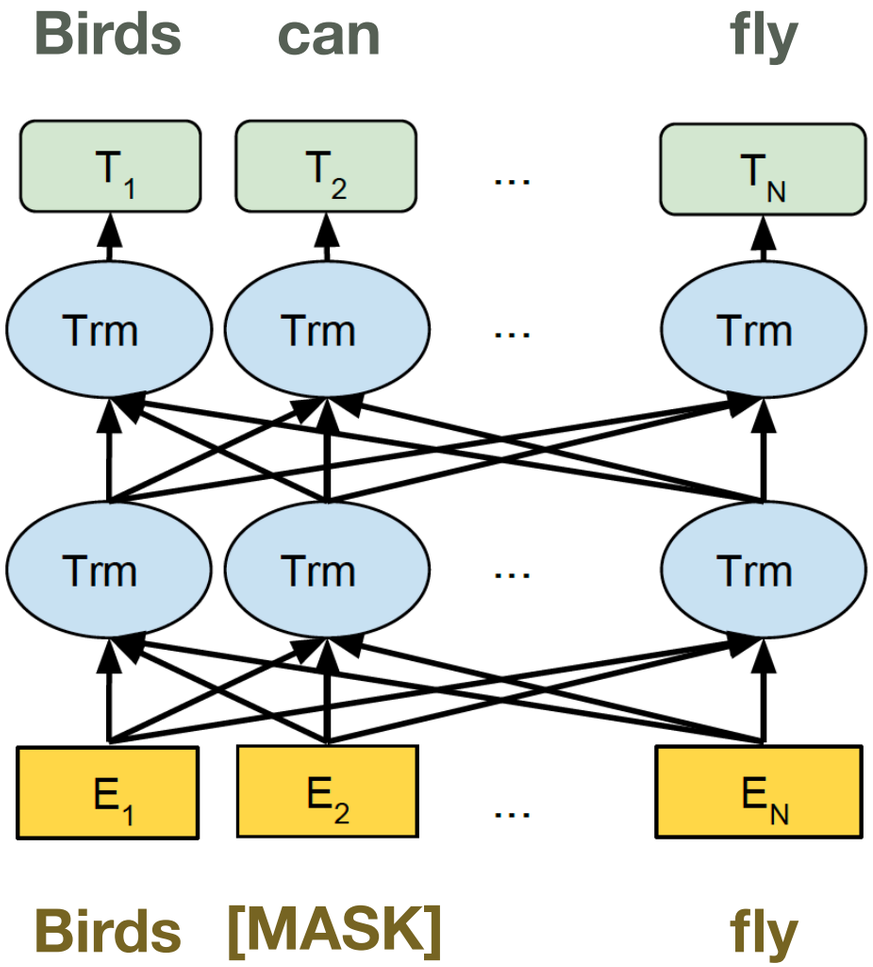
Masked language modeling



(In practice some of the input words may be subwords!)

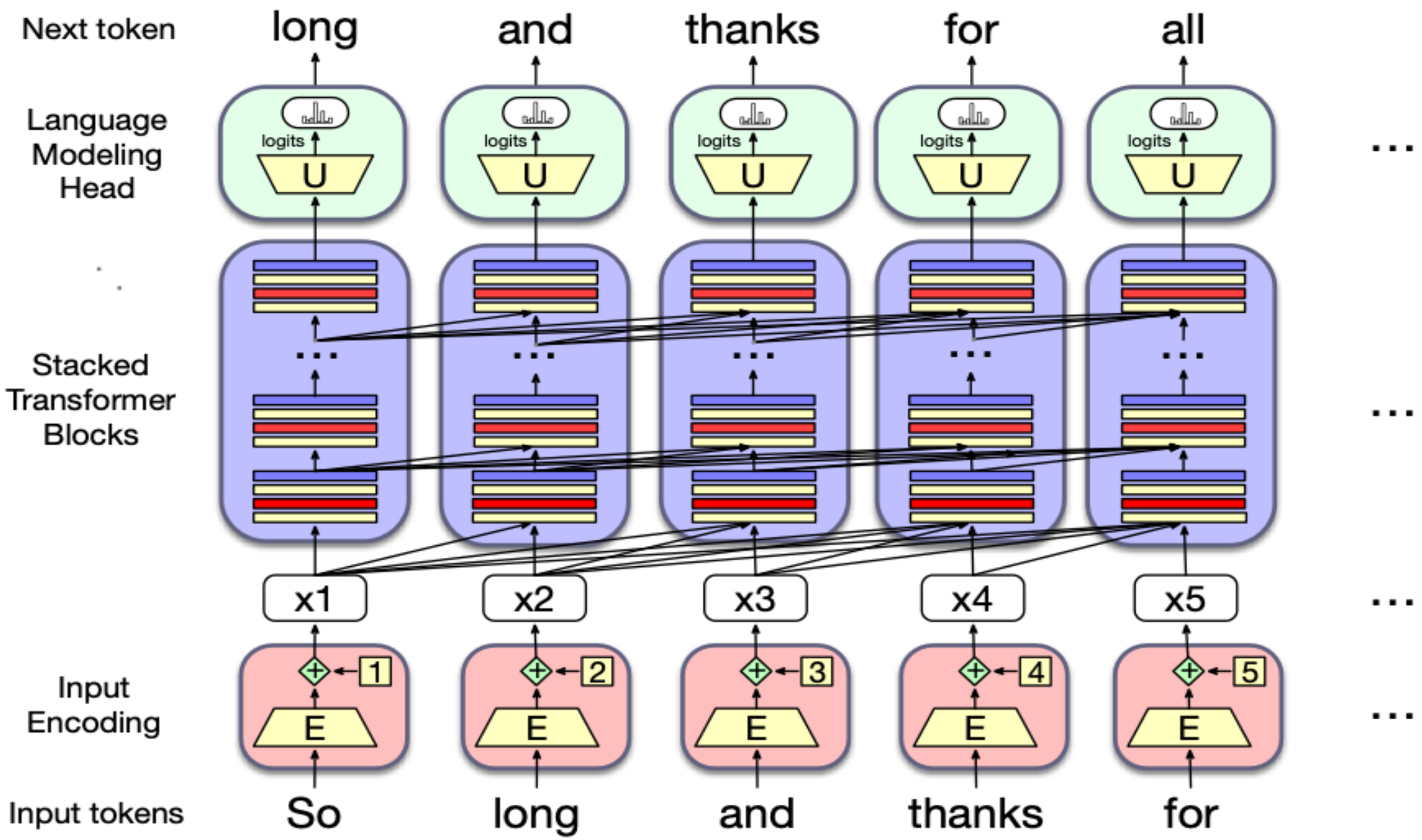
BERT: encoder-only transformer with a masked language modeling objective

- Doesn't obviously give us a probability distribution over sequences of words, so it's an extension of the term "language model"



(Devlin et al., 2019)

Decoder-only (“causal”) transformer as an unconditioned language model



Do we even need to fine-tune? Language Models as Unsupervised Multitask Learners

- Key hypothesis: the inputs and output for a lot of tasks sometimes naturally occur in a training corpus

- Given a large enough corpus, a powerful enough statistical learner will implicitly learn to perform the tasks just to reduce perplexity

- Amazingly, this works for a lot of tasks

(Radford et al 2019)

Natural demonstrations of French to English translations

”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**”.

(Radford et al 2019)

Do we even need to fine-tune? Language Models as Unsupervised Multitask Learners

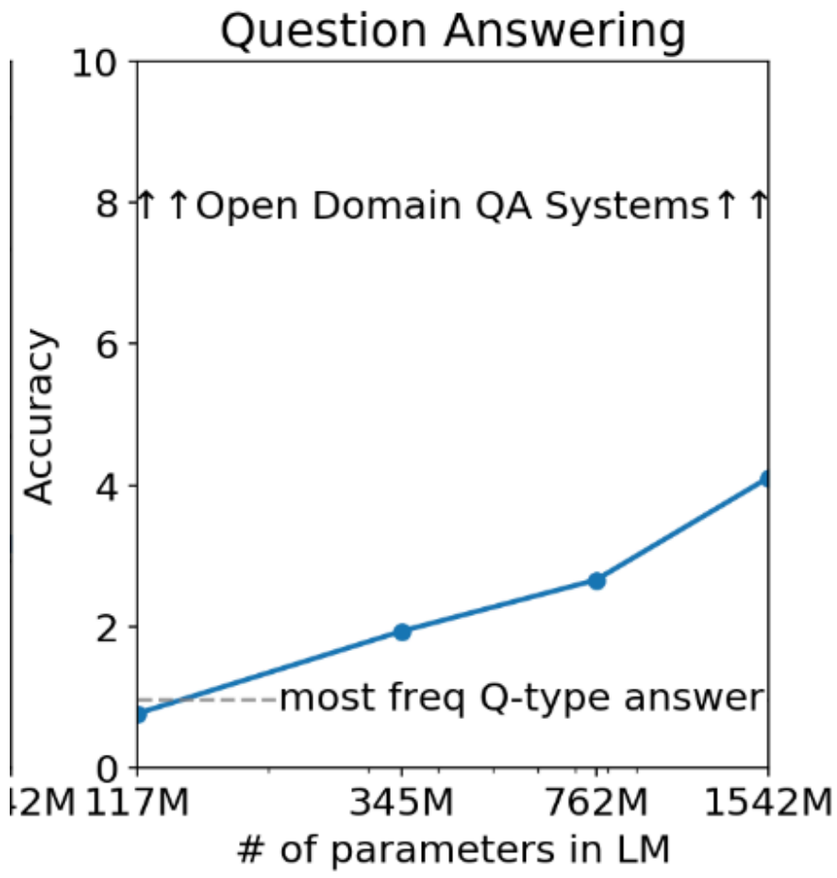
- No formal separation between encoder and decoder: we provide the input through teacher forcing (this is called the “prompt”)

- E.g., to answer questions about the document, we provide the document, the question, append “A:”, then generate

- To perform summarization, we append “TL;DR:” to the input (!), then generate

(Radford et al 2019)

Do we even need to fine-tune? Language Models as Unsupervised Multitask Learners



(Radford et al 2019)

“In-context learning”

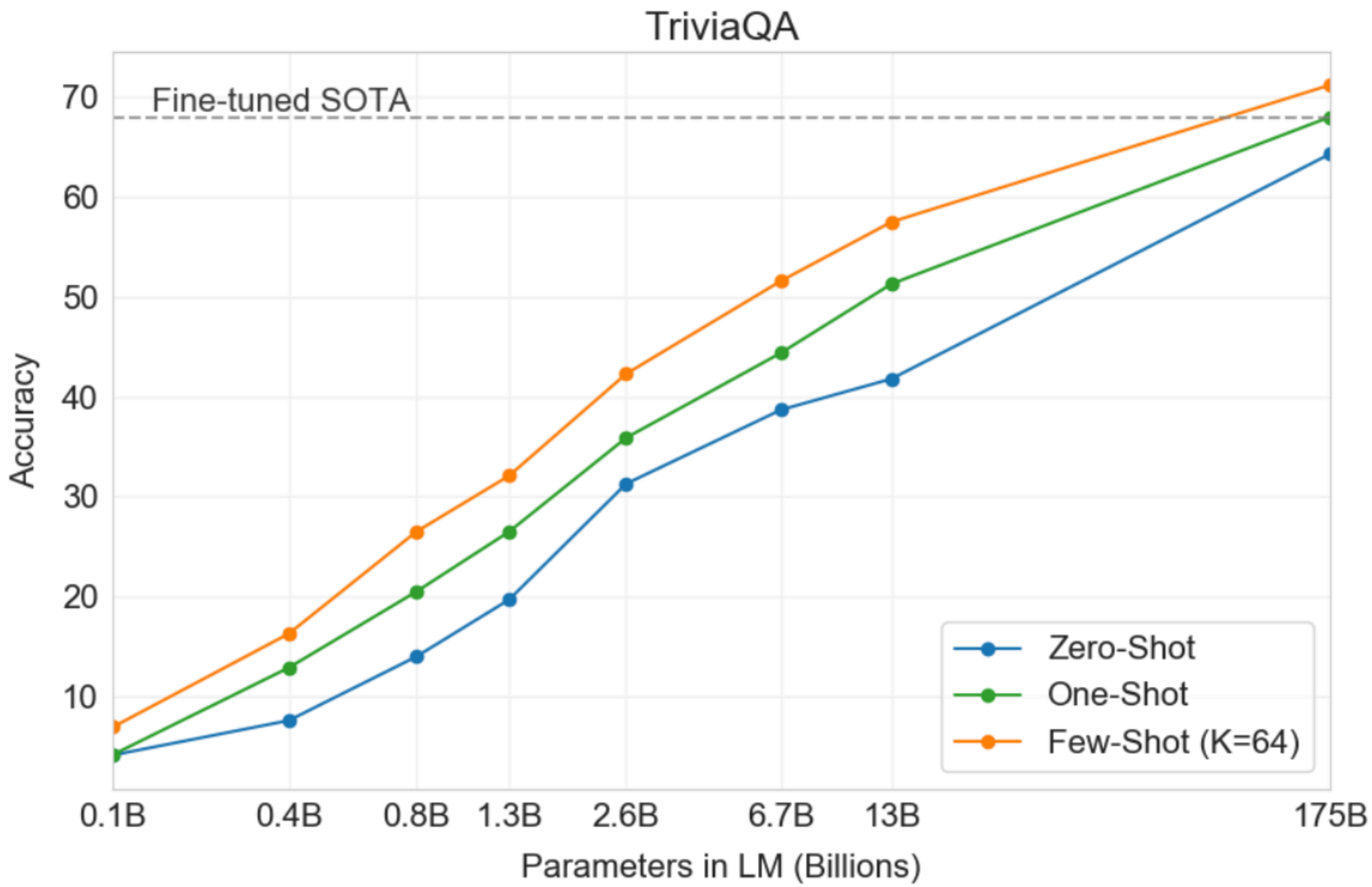
- Performance can be improved if we provide some demonstrations of input and output pairs in context (e.g. eight French sentences and their English translations)

- This isn’t learning in the sense that we do not compute gradients and do not change the model parameters

- But this can put the model’s activations in a place where they are somewhat more likely to get the model to perform the task

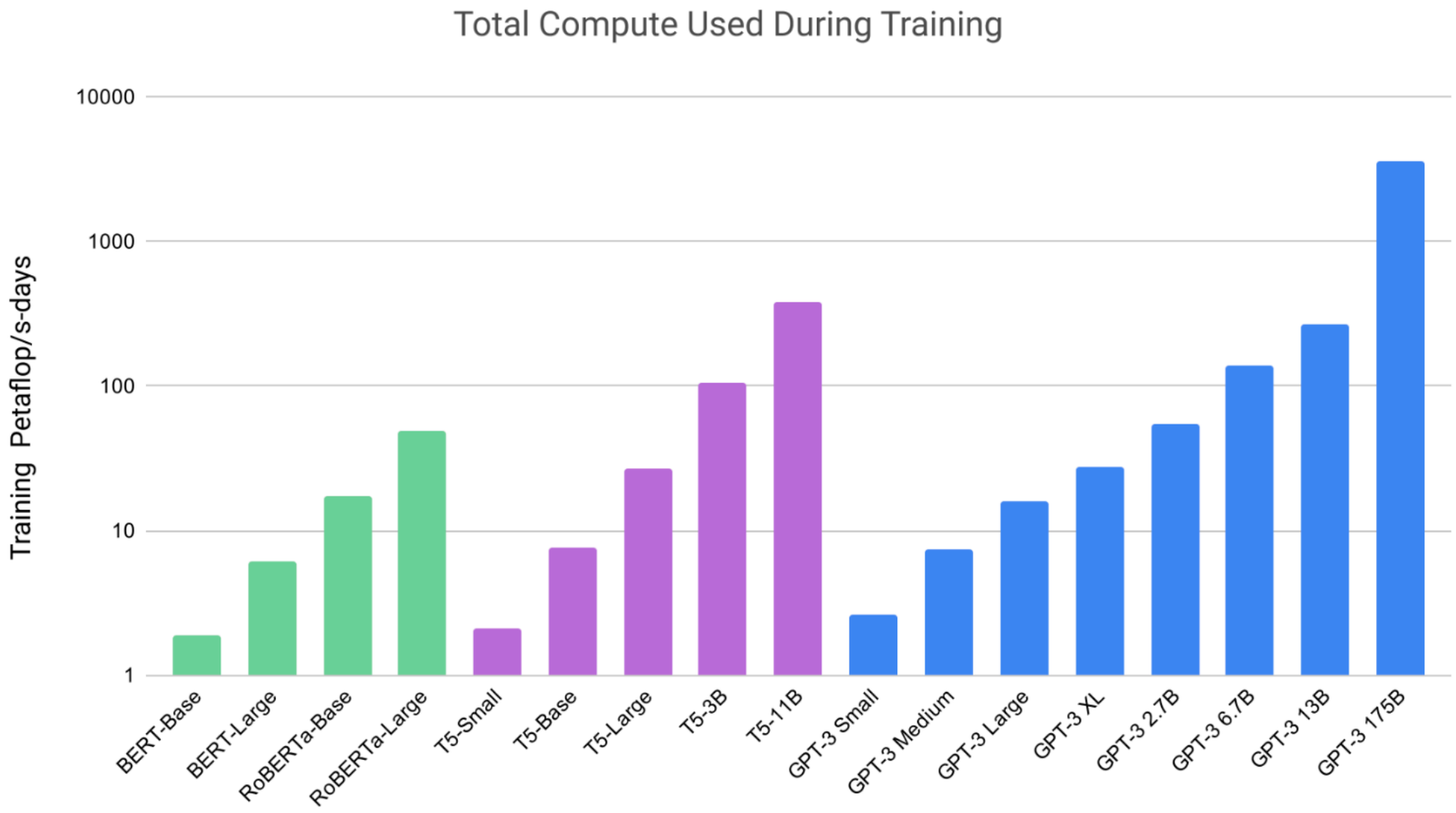
(Brown et al 2020)

“In-context learning”



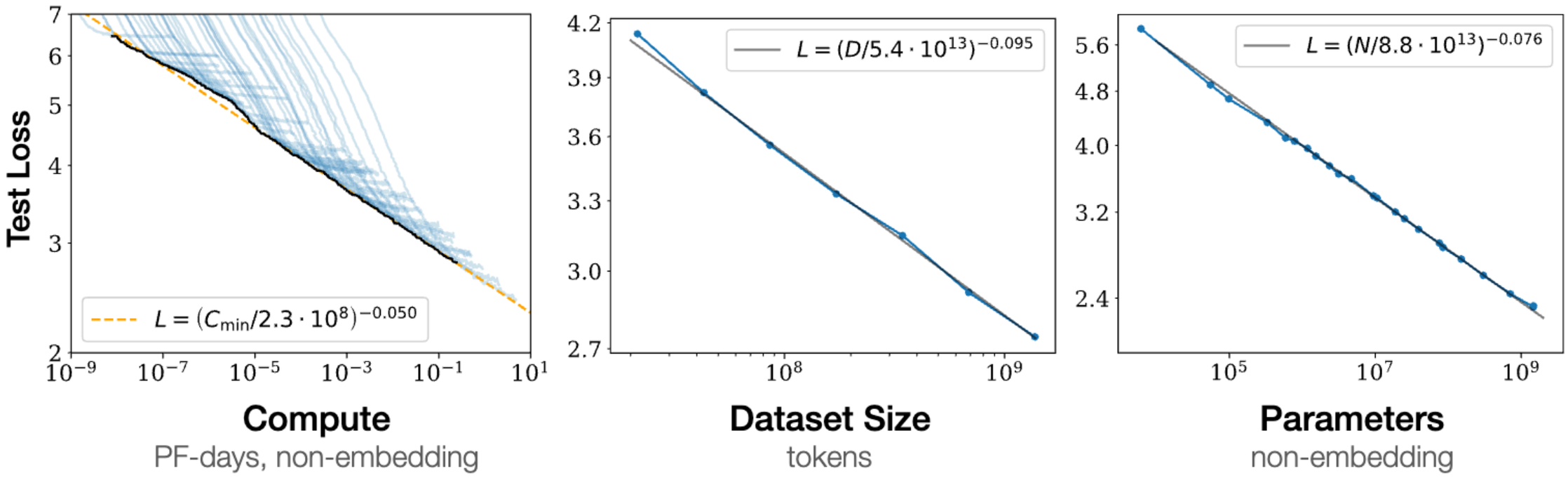
(Brown et al 2020)

Scaling



(Brown et al 2020)

Language modeling loss improves smoothly with the amount of compute

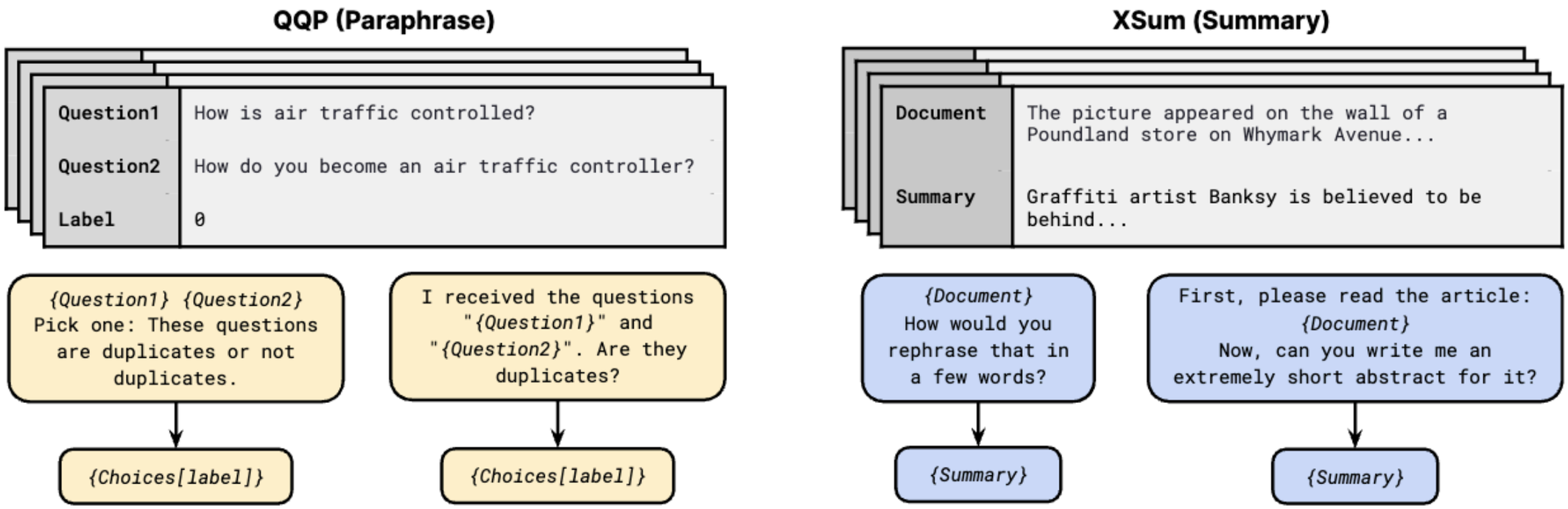


(Kaplan et al 2020)

Reintroducing fine-tuning after all

- Self-supervised learning on a huge corpus (pre-training) can, amazingly, induce some task-solving abilities
- But we do still have the datasets we used for supervised learning, or to fine-tune BERT, why not use them?
- Let's combine the two!
- This is called "supervised fine-tuning" (SFT) or "instruction tuning"
- After fine-tuning the model can generalize to novel instructions!

Supervised fine-tuning



(Sanh et al., 2022)

Supervised fine-tuning

- LLMs have a lot of parameters, so this can get expensive
- Often we can get away with fine-tuning only a small subset of parameters (“parameter-efficient fine-tuning”, e.g. LoRA, Hu et al 2021)

Supervised fine-tuning

- We can get creative with the prompts: they don't need to be traditional NLP tasks

- E.g., we can pay writers to write haikus for us

- We can even use LLM-generated synthetic data (one recent example in the next few slides)

Learning from preferences

- For a particular input, often there's no one correct output, but some outputs are better than others
- Some properties of the input are very important to preserve in the output, others don't matter as much
- We can generate multiple outputs for the input (e.g., from different models, or different sampling methods), and annotators to assign a score
- In practice it's easier for annotators to compare two outputs to each other; we can derive a scalar "goodness" measure from this head-to-head comparisons

Learning from preferences

- We only get the reward at the end of the language model's generation: need to use reinforcement learning

- We can have another language model provide the preferences, automating the whole process!