
Intelligent Agents

1d-CNNs LSTMs ELMo Transformers BERT GPT

Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme



Acknowledgements

- Some slides are based on
 - CS546: Machine Learning in NLP (Spring 2020)
 - <http://courses.engr.illinois.edu/cs546/>
 - Julia Hockenmaier <http://juliahr.cs.illinois.edu>
 - RNs, LSTMs, ELMo, Transformers
 - Machine Learning (Spring 2020)
 - http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML20.html
 - 李宏毅 (Hung-yi Lee) <http://speech.ee.ntu.edu.tw/~tlkagk/>
 - ELMo, BERT: [http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2019/Lecture/BERT%20\(v3\).pdf](http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2019/Lecture/BERT%20(v3).pdf)
- Respective sources are indicated in the gray line at the bottom
- Slides have been modified
 - All errors are mine

Recap: Convolution

Input image



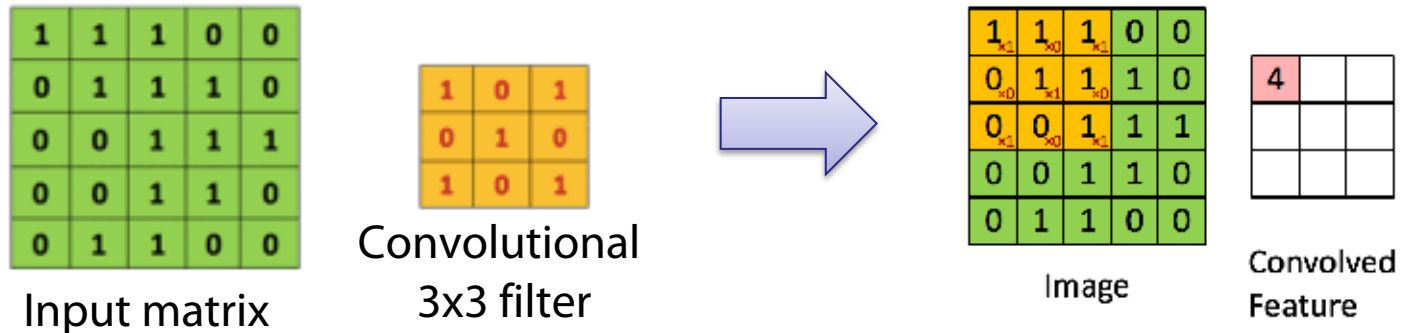
Convolution
Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map



Recap: Convolutional Networks (CNs)



Main ConvNet idea for text:

Compute vectors for n-grams and group them afterwards

Example: “this takes too long” compute vectors for:

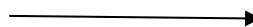
This takes, takes too, too long, this takes too, takes too long, this takes too long

Recap: ConvNets (CNs)

Feature Map

6	4	8	5
5	4	5	8
3	6	7	7
7	9	7	2

max pool
2x2 filters
and stride 2



Max-Pooling

Dimension reduction

Main ConvNet idea for text:

Compute vectors for n-grams and **group them afterwards**

<https://shafeentejani.github.io/assets/images/pooling.gif>

1d-CNNs for text

Text is a (variable-length) sequence of words (word vectors)

We can use a 1d-CNN to slide a window of n tokens across:

— filter size $n = 3$, stride = 1, no padding

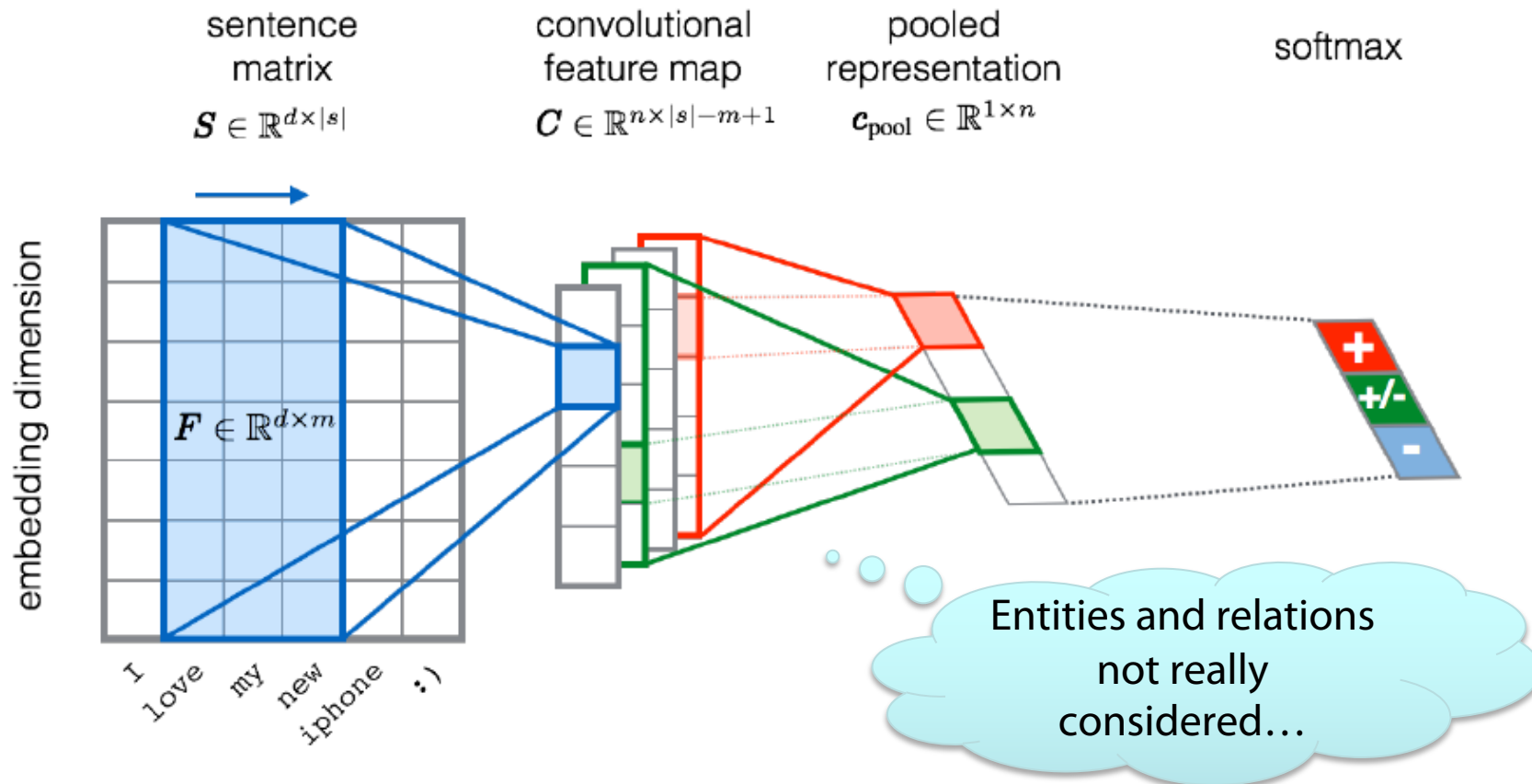
The quick brown fox jumps over the lazy dog
The **quick brown fox** jumps over the lazy dog
The quick **brown fox jumps** over the lazy dog
The quick brown **fox jumps over** the lazy dog
The quick brown fox **jumps over the** lazy dog
The quick brown fox jumps **over the lazy** dog

— filter size $n = 2$, stride = 2, no padding:

The quick brown fox jumps over the lazy dog
The quick **brown fox** jumps over the lazy dog
The quick brown fox **jumps over** the lazy dog
The quick brown fox jumps over **the lazy** dog

CNNs (w/ ReLU and maxpool) can be used for classifying (parts of) the text

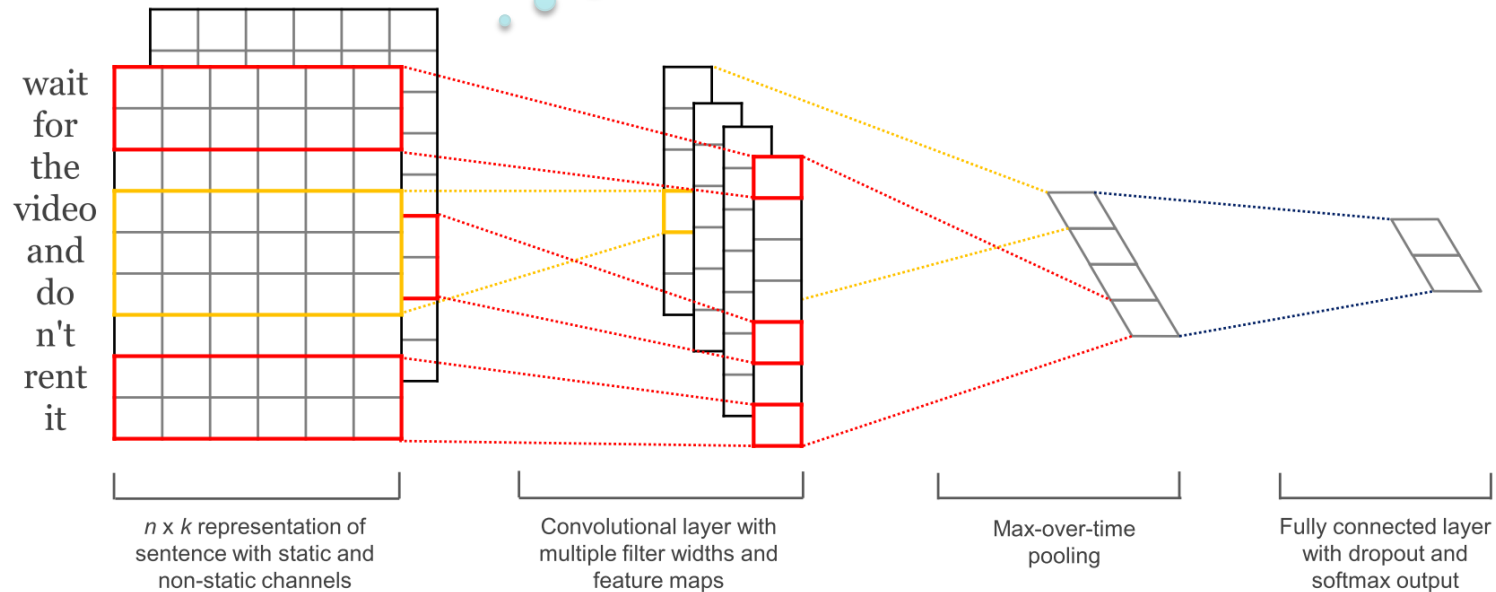
CNNs for sentiment analysis



Severyn, Aliaksei, and Alessandro Moschitti. "UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification." *SemEval@NAACL-HLT*. 2015.

CNNs for sentence/text classification

Pretrained and task-specific embeddings: Multiple Channels



Kim, Y. "Convolutional Neural Networks for Sentence Classification", EMNLP (2014)

sliding over 3, 4 or 5 words at a time

Static = pre-trained, non-static = task-specific

Fasttext (<https://fasttext.cc>)

- Library for word embeddings and text classification
 - static word embeddings and ngram features
 - that get averaged together in one hidden layer
 - hierarchical softmax output over class labels
- Enriching word vectors with subword information
 - Skipgram model where each word is a sum of character ngram embeddings and its own embedding
 - Each word is deterministically mapped to ngrams

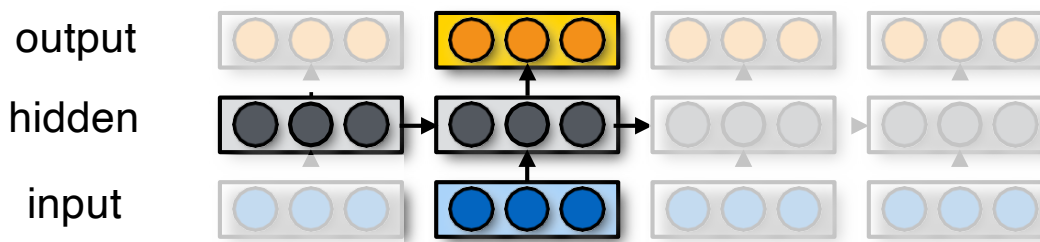
Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, Volume 5. 135-146. **2017**.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, Bag of Tricks for Efficient Text Classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. 427-431. **2017**.

Alon Jacovi, Oren Sar Shalom, Yoav Goldberg. Understanding Convolutional Neural Networks for Text Classification. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. **2018**.

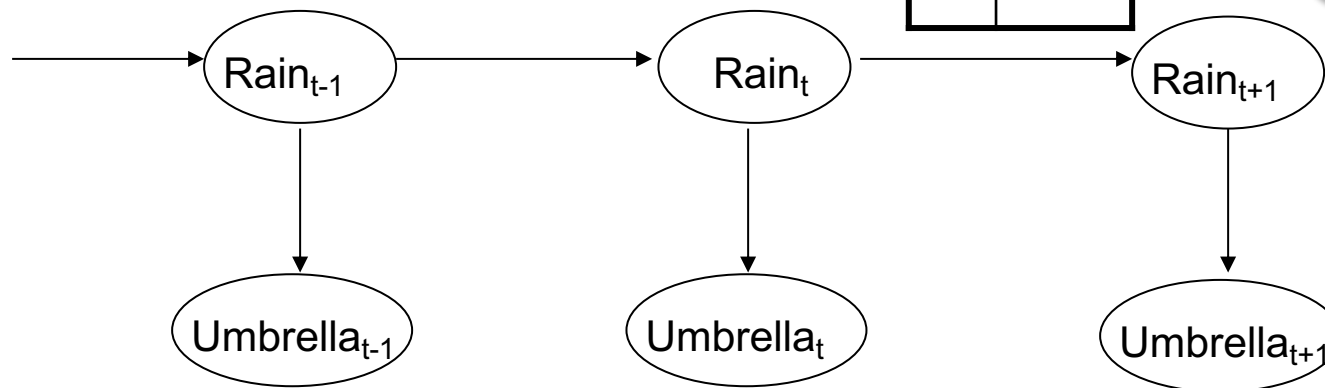
Recursive Networks – Or: Copying the Pattern

- Basic computational network copied per time slice
- Input: previous hidden state, output: next hidden state



Computational model

- Compare with HMM (or CRF):

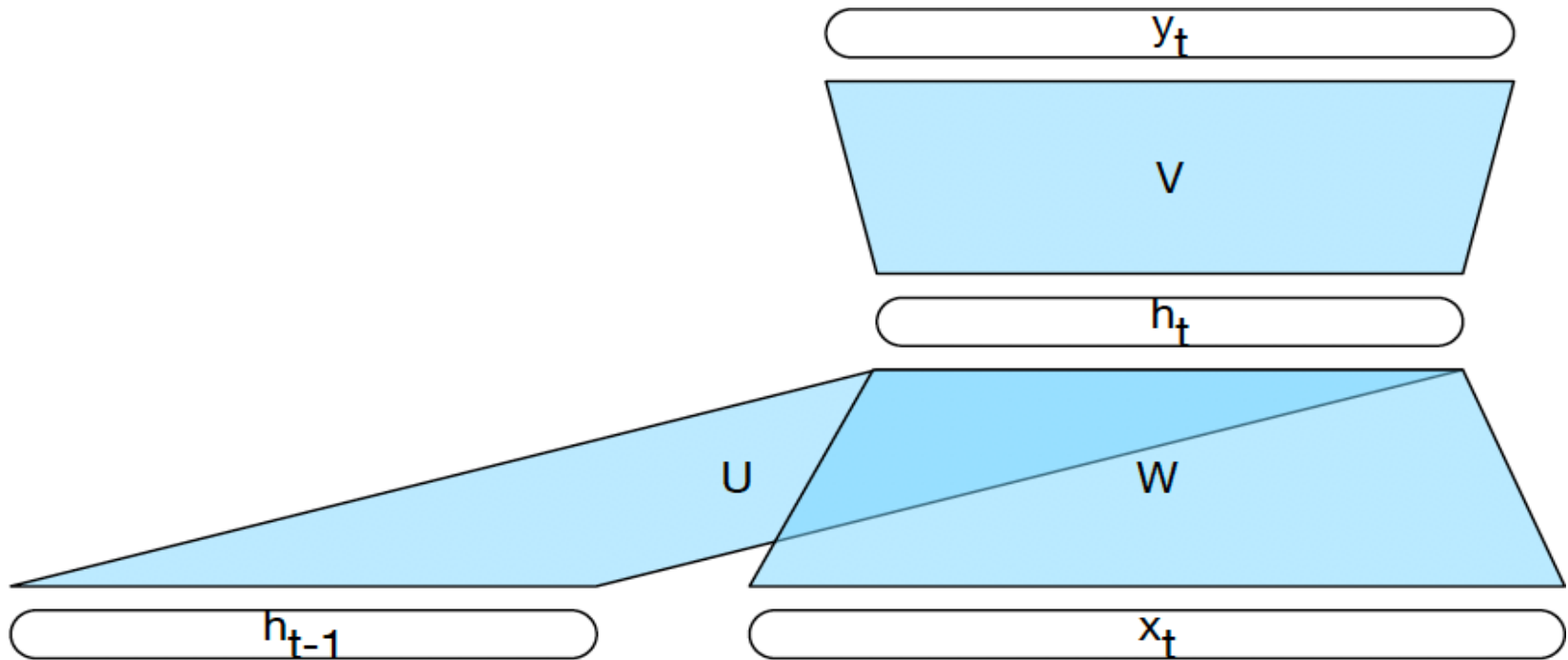


R_{t-1}	$P(R_t R_{t-1})$
T	0.7
F	0.3

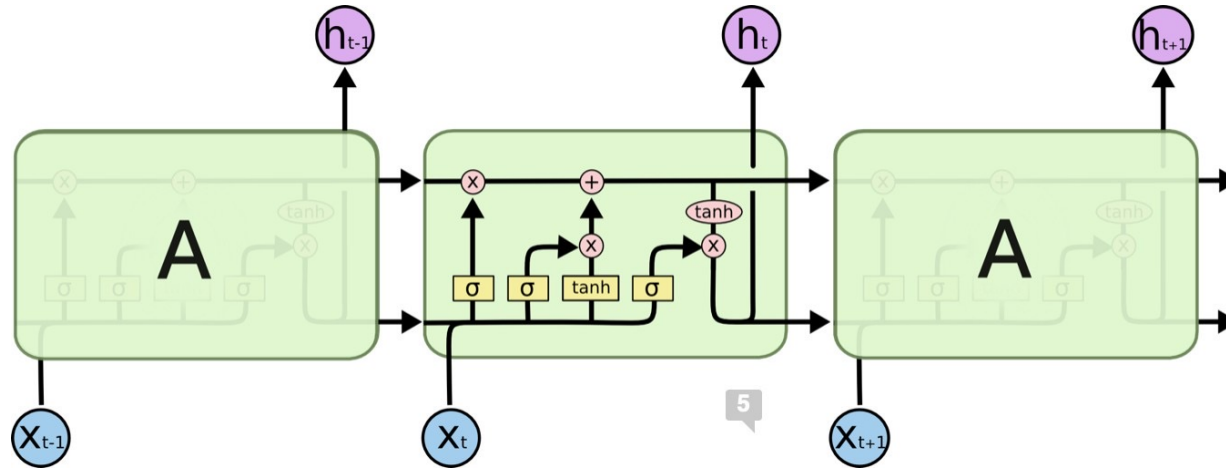
R_t	$P(U_t R_t)$
T	0.9
F	0.2


Declarative model (generative)

Computing the Hidden State




Long Short Term Memory Networks (LSTMs)




Neural Network
Layer


Pointwise
Operation


Vector
Transfer


Concatenate


Copy

Recap: Activation Functions

Sigmoid (logistic function):

$$\sigma(x) = 1/(1 + e^{-x})$$

Returns values bound above and below in the $0,1$ range

Hyperbolic tangent:

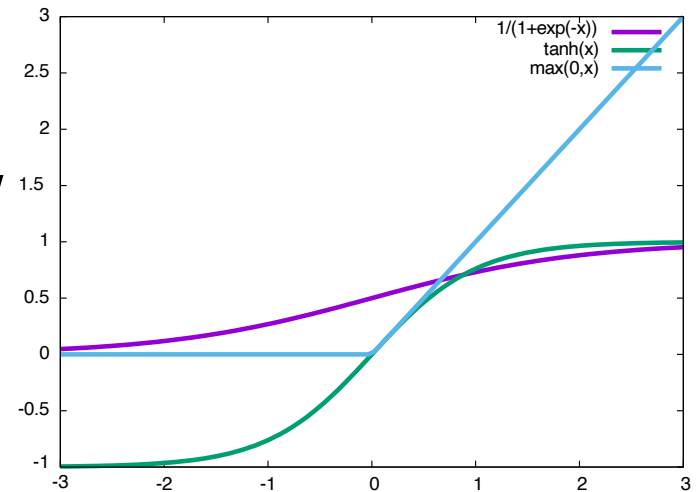
$$\tanh(x) = (e^{2x}-1)/(e^{2x}+1)$$

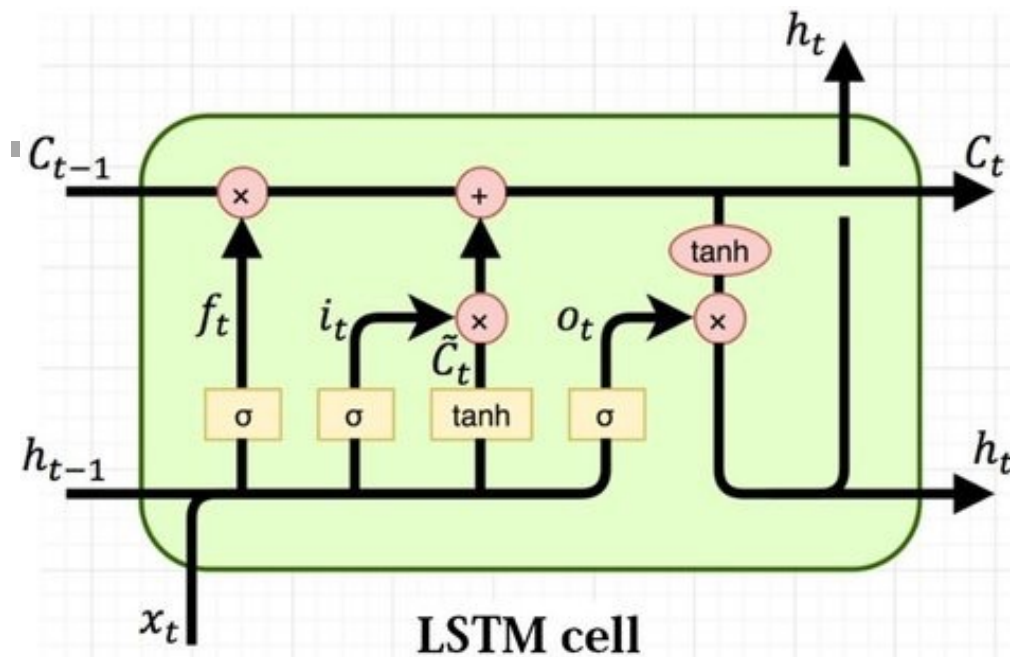
Returns values bound above and below in the $-1,+1$ range

Rectified Linear Unit:

$$\text{ReLU}(x) = \max(0, x)$$

Returns values bound below in the $0, +\infty$ range





$$i_t = \sigma(x_t U^i + h_{t-1} W^i)$$

$$f_t = \sigma(x_t U^f + h_{t-1} W^f)$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o)$$

$$\tilde{C}_t = \tanh(x_t U^g + h_{t-1} W^g)$$

$$C_t = \sigma(f_t * C_{t-1} + i_t * \tilde{C}_t)$$

$$h_t = \tanh(C_t) * o_t$$

At time t , the LSTM cell reads in

- a c -dimensional previous **cell state vector** \mathbf{c}_{t-1}
- an h -dimensional previous **hidden state vector** \mathbf{h}_{t-1}
- a d -dimensional current **input vector**

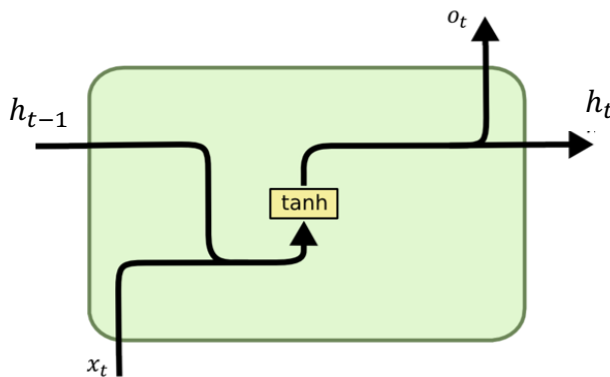
At time t , the LSTM cell **returns**

- a c -dimensional previous **cell state vector** \mathbf{c}_t
- an h -dimensional previous **hidden state vector** \mathbf{h}_t
(which may also be passed to an output layer)

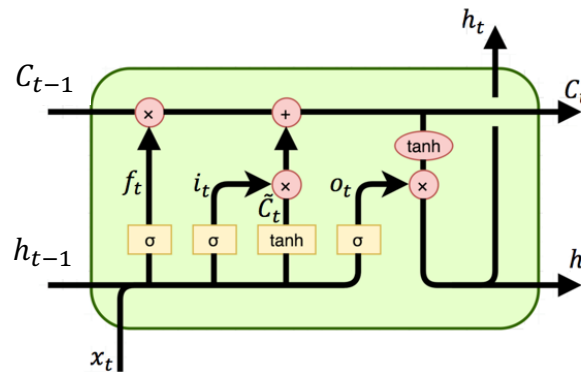
Repetitive Variants: LSTMs, GRUs

- **Long Short Term Memory** networks (LSTMs) are RNs with a more complex architecture to combine the last hidden state with the current input.
- **Gated Recurrent Units** (GRUs) are a simplification of LSTMs
- Both contain “**gates**” to control how much of the input or past hidden state to forget or remember

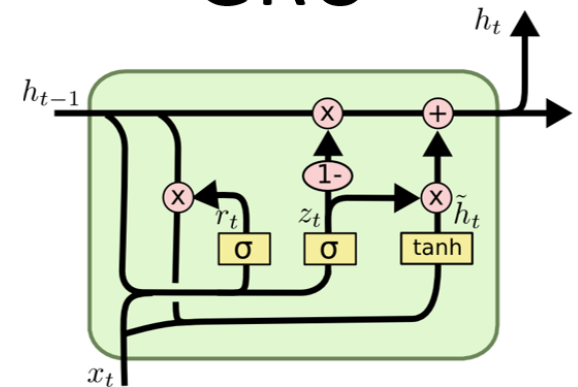
RN



LSTM



GRU

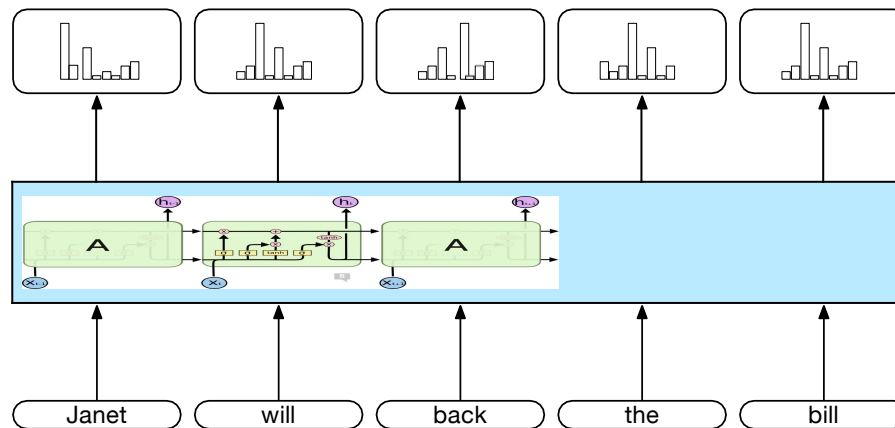


Gates

- A gate performs element-wise multiplication of
 - the output of a d -dimensional sigmoid layer (all elements between 0 and 1), and
 - a d -dimensional input vector
- Result: a d -dimensional output vector which is like the input, except some dimensions have been (partially) “forgotten”

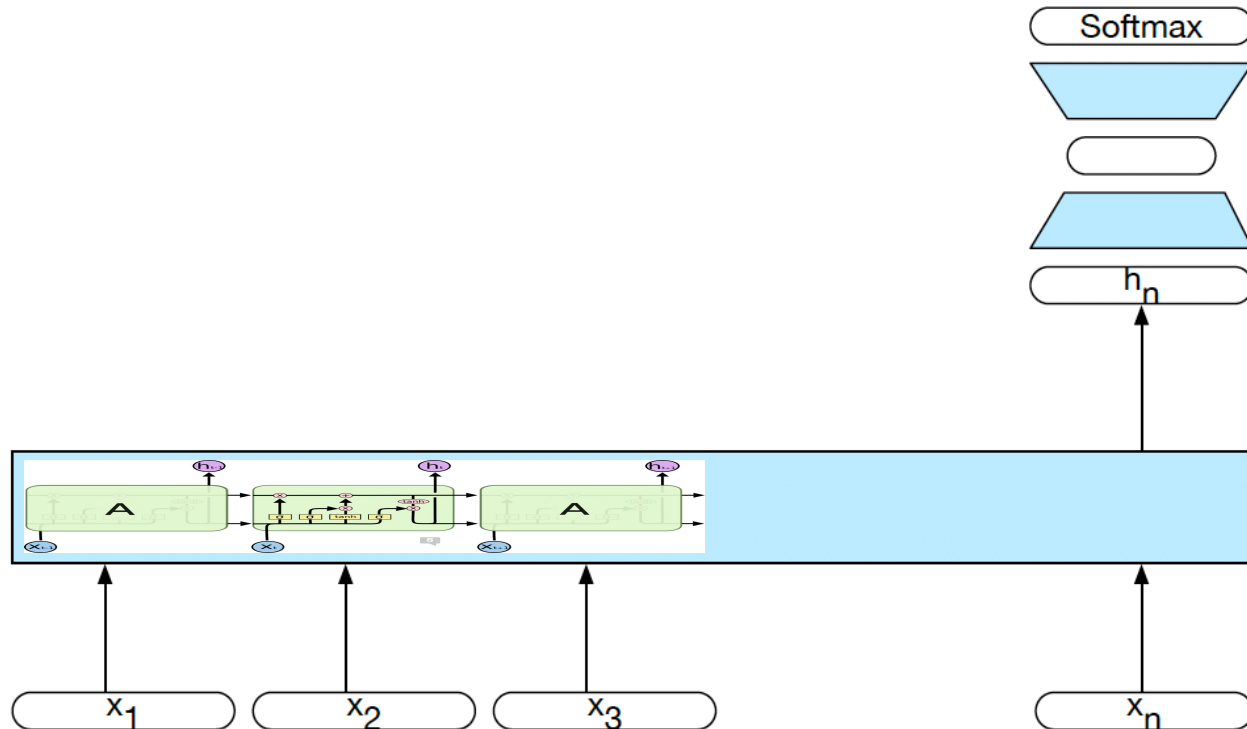
Basic RNs for Sequence Labeling

Each time step has a distribution over output classes



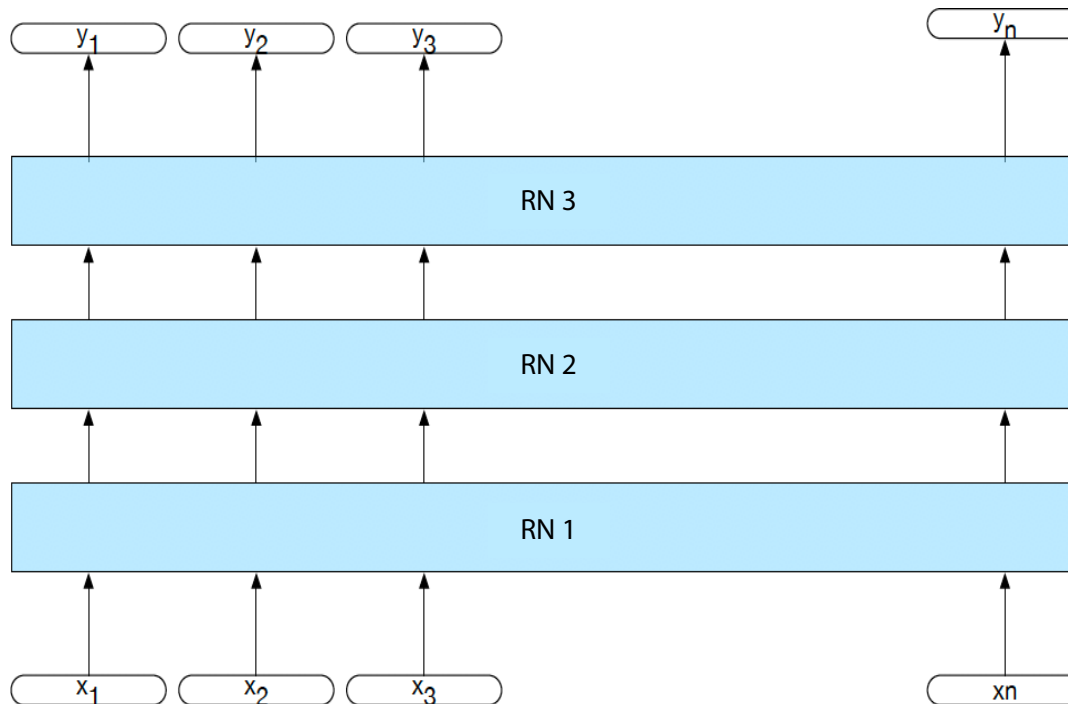
Extension: add a HMM/CRF layer to capture dependencies among labels of adjacent tokens.

RNs for Sequence Classification

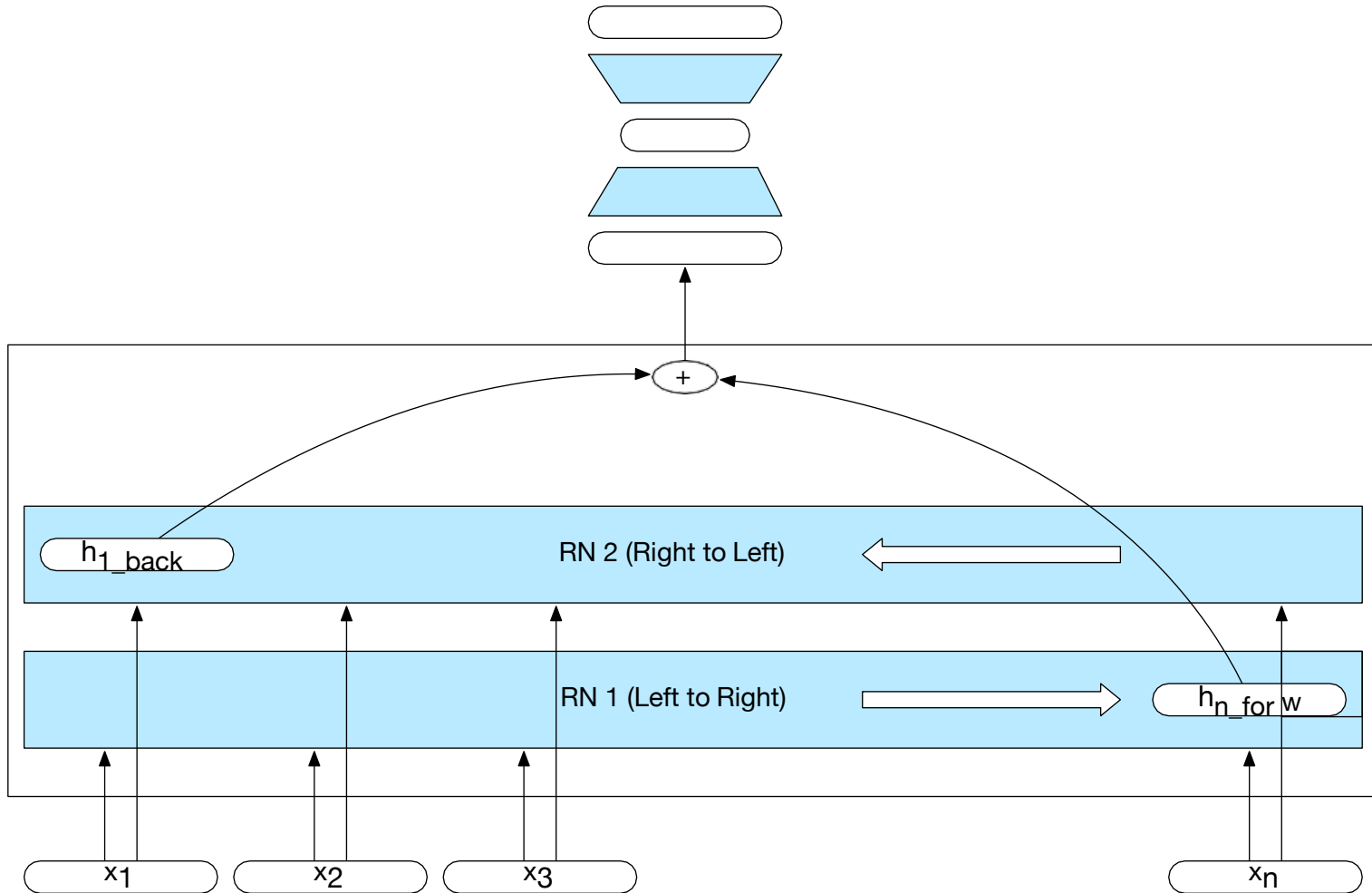


Stacked RNs

We can create an RN that has “vertical” depth (at each time step) by stacking multiple RNs:

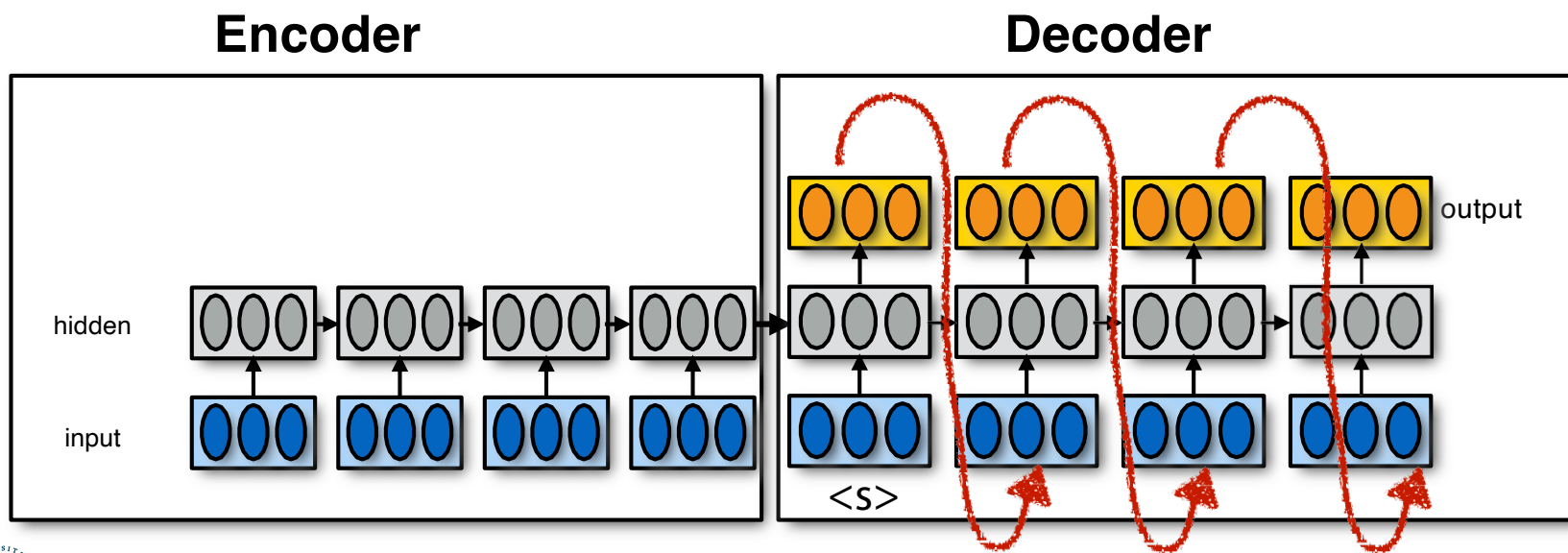


Bidirectional RNNs



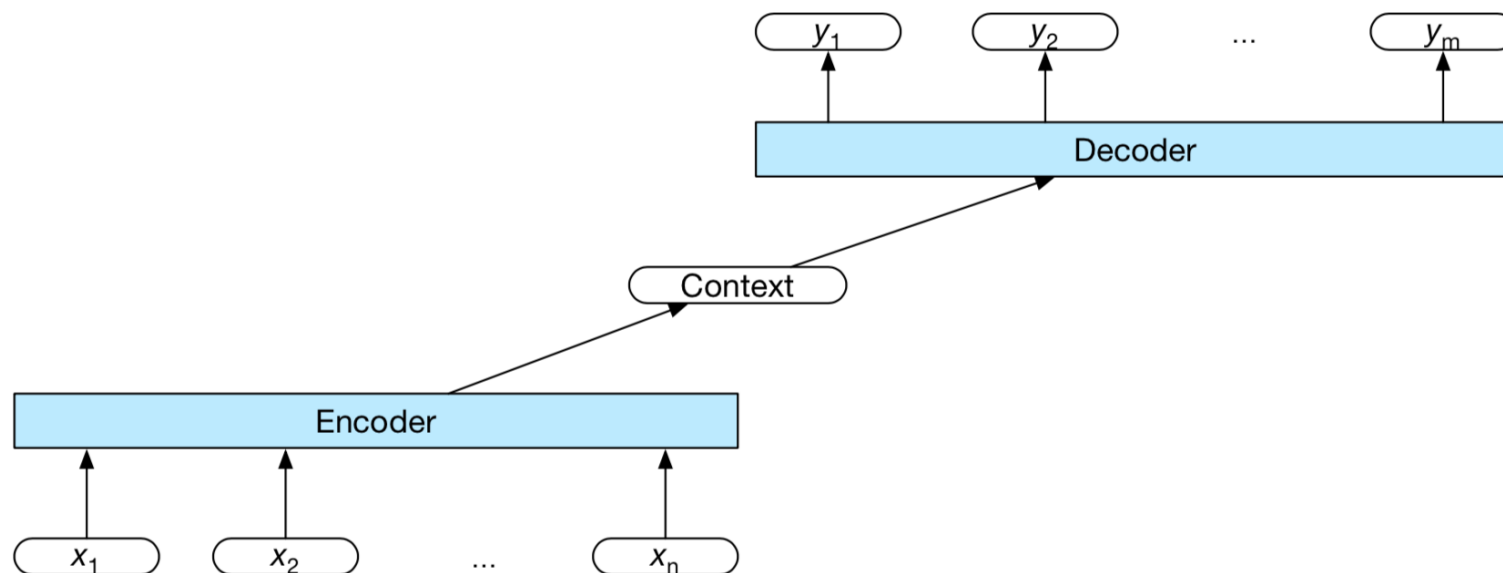
Encoder-Decoder (seq2seq) model

- Task: Read an input sequence and return an output sequence
 - Machine translation: translate source into target language
 - Dialog system/chatbot: generate a response
- Reading the input sequence: RN Encoder
- Generating the output sequence: RN Decoder



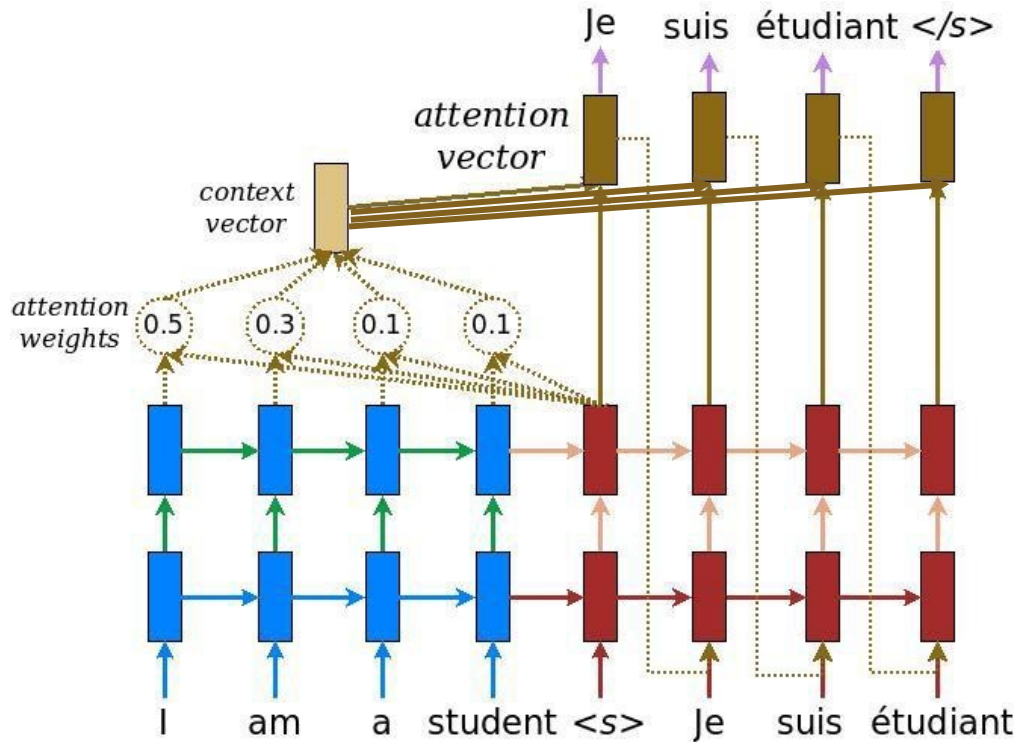
A More General View of seq2seq

In general, any function over the encoder's output can be used as a representation of the context we want to condition the decoder on.



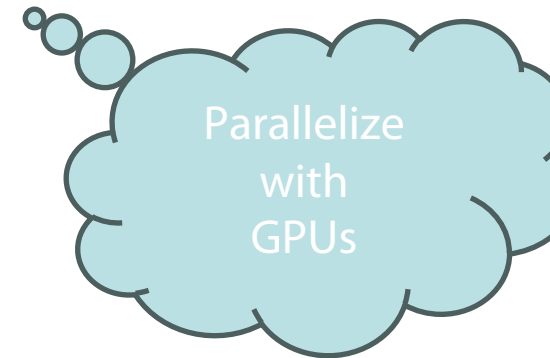
We can feed the context in at any time step during decoding (not just at the beginning).

Attention Mechanisms



Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio:
Neural Machine Translation by Jointly Learning to Align and
Translate. ICLR **2015**

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit,
Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin.
2017. Attention is all you need. In Proceedings of the 31st
International Conference on Neural Information Processing
Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA,
6000–6010. **2017**



Embeddings from Language Models

Replace static embeddings (lexicon lookup) with **context-dependent embeddings** (produced by a deep language model)

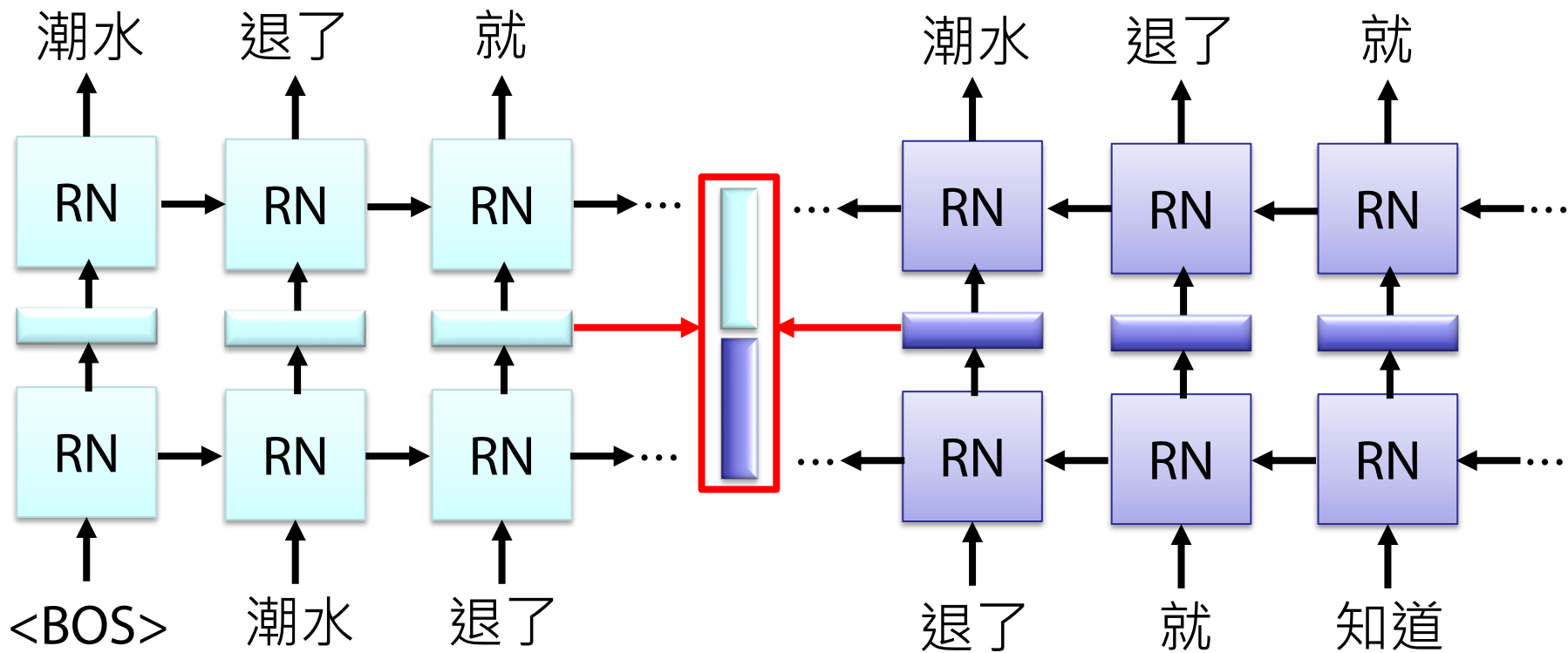
=> Each token's representation is a function of the entire input sentence, computed by a deep **(multi-layer) bidirectional language model**

=> Return for each token a **(task-dependent) linear combination of its representation across layers.**

=> Different layers capture different information

Embeddings from Language Model (ELMO)

- RN-based language models (trained from lots of sentences) e.g., given “潮水 退了 就 知道 誰 沒穿 褲子”

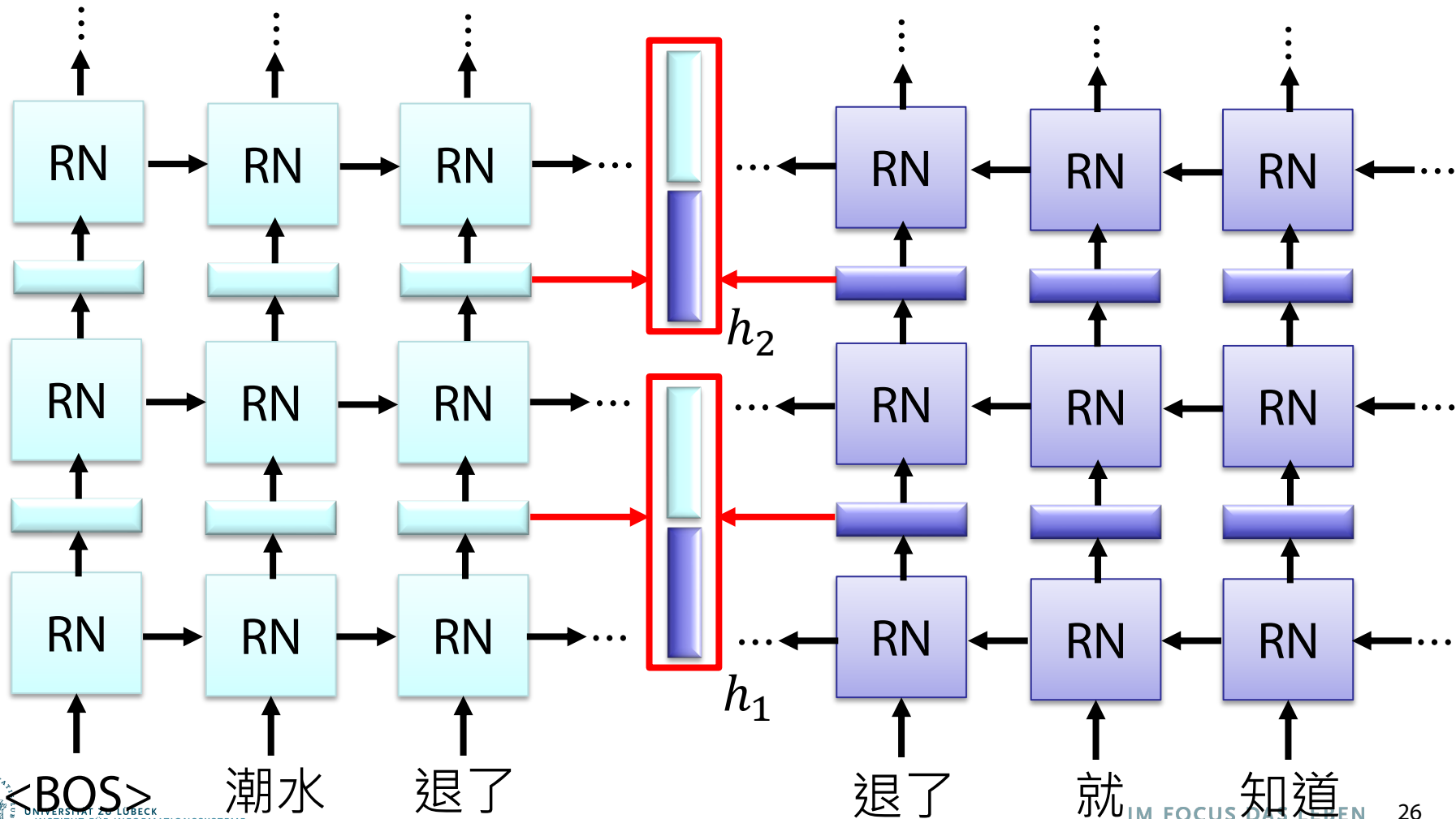


潮水 退了 就 知道 誰 沒穿 褲子 = When the tide goes out, you know who's not wearing pants.

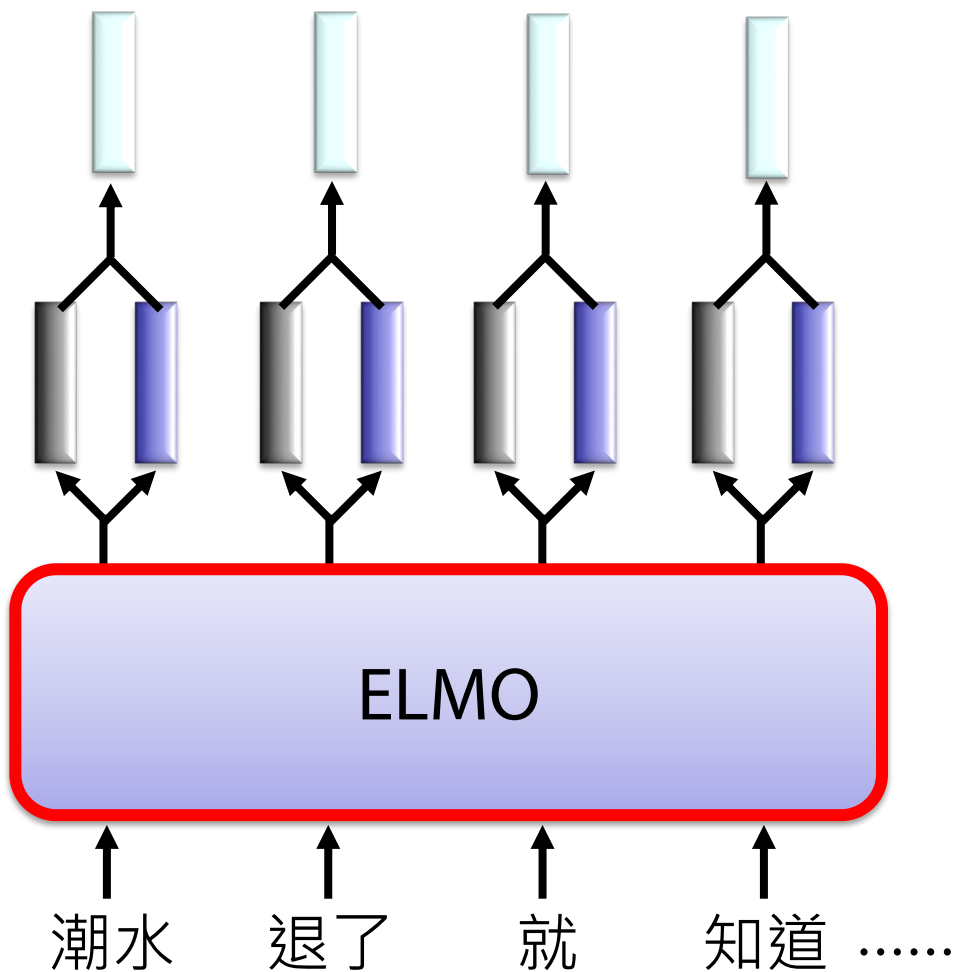
ELMO

Each layer in deep LSTM can generate a latent representation.

Which one should we use???



ELMO

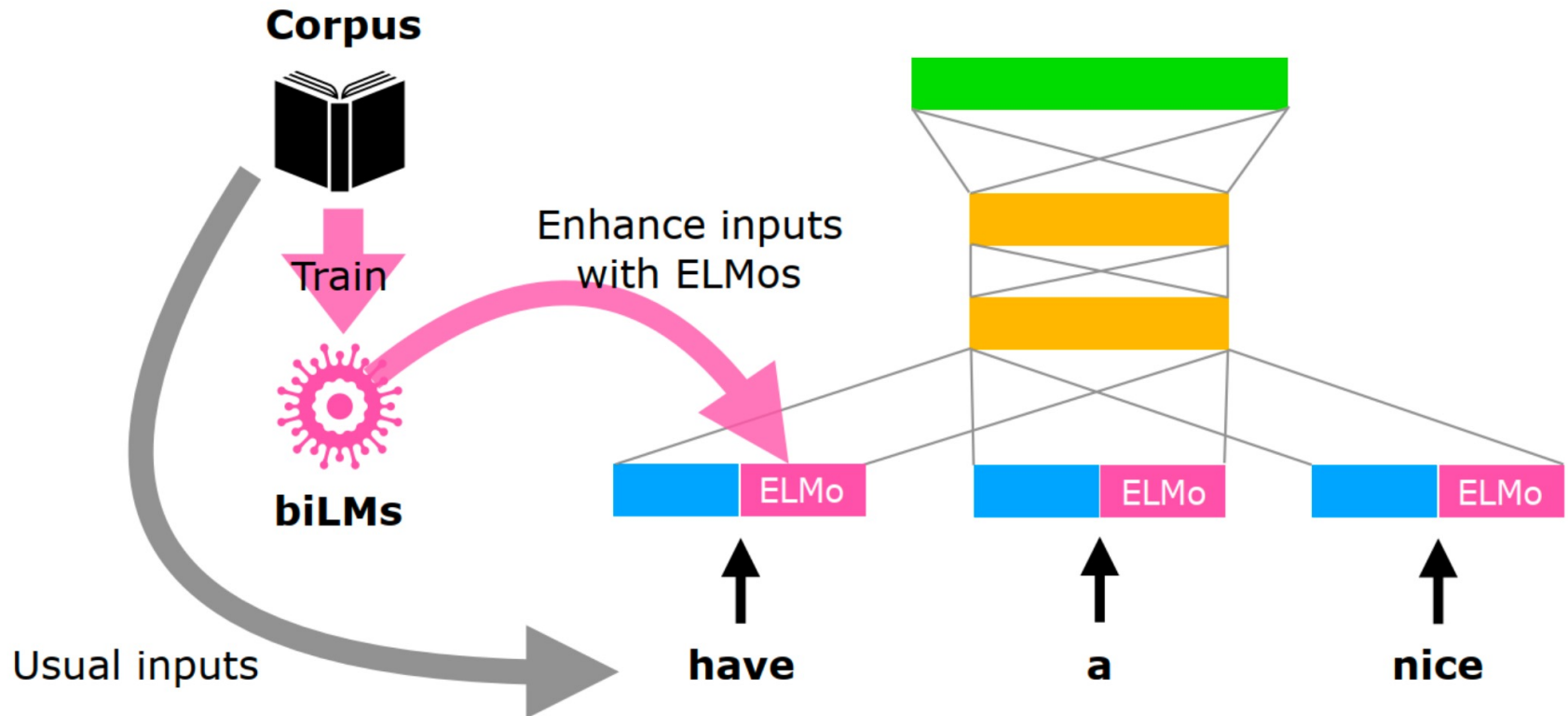


High computational effort, word2vec to the rescue?

$$\text{Light Blue Bar} = \alpha_1 \text{Grey Bar} + \alpha_2 \text{Blue Bar}$$

Learned with the down-stream tasks

Integrate ELMos into other embeddings



Tricks: Subtoken Encoding

Byte Pair Encoding (BPE)

Word embedding sometimes is too high level, pure character embedding too low level. For example, if we have learned

old older oldest

We might also wish the computer to infer

smart smarter smartest

But at the whole word level, this might not be so direct. Thus, the idea is to break the words up into pieces like er, est, and embed frequent fragments of words.

GPT adapts this BPE scheme.

Tricks: Subtoken Encoding

Byte Pair Encoding (BPE)

GPT uses BPE scheme. The subwords are calculated by:

1. Split word to sequence of characters (add `</w>` char)
2. Joining the highest frequency pattern.
3. Keep doing step 2, until it hits the pre-defined maximum number of subwords or iterations.

Example (5, 2, 6, 3 are number of occurrences)

{`low </w>`:5, `lower </w>`:2, `newest </w>`:6, `widest </w>`:3 }

{`low </w>`:5, `lower </w>`:2, `newest </w>`:6, `widest </w>`:3 }

{`low </w>`:5, `lower </w>`:2, `newest </w>`:6, `widest </w>`:3 } ("est" freq. 9)

{`low </w>`:5, `lower </w>`:2, `newest </w>`:6, `widest </w>`:3 } ("lo" freq 7)

.....

The end of the neural AI era: Postneural AI

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin* †
illia.polosukhin@gmail.com

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), 6000–6010. **2017**.

Self-attention

q : query (to match others)

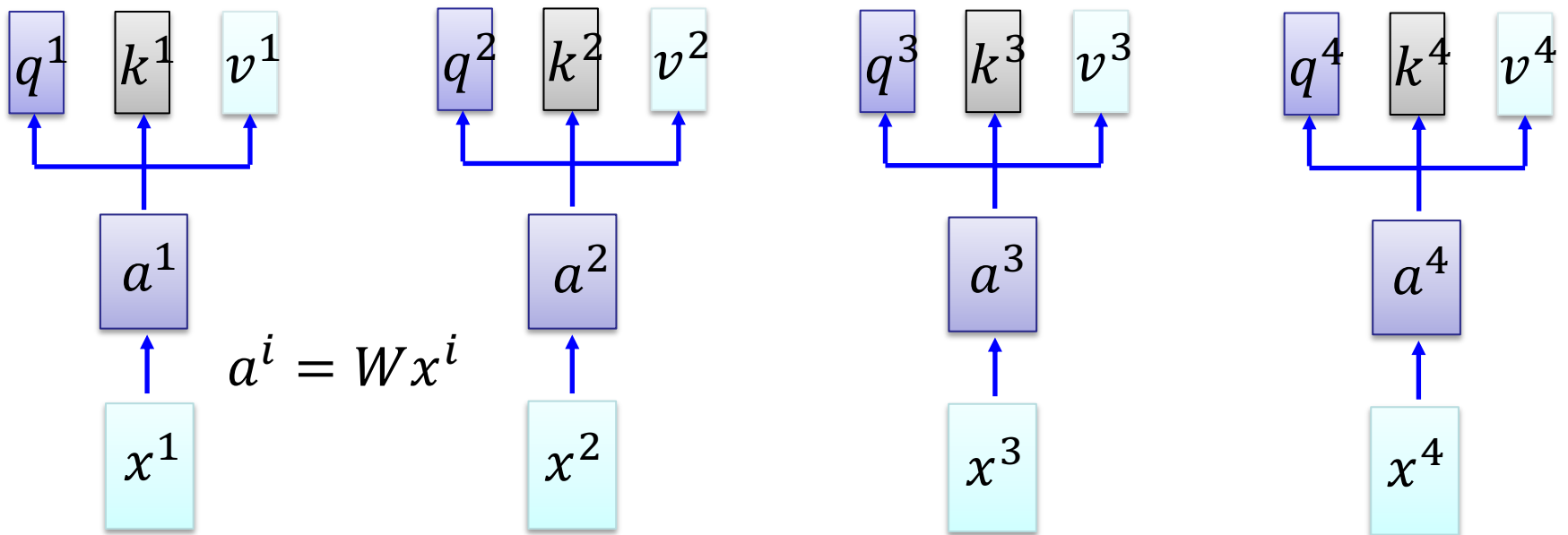
$$q^i = W^q a^i$$

k : key (to be matched)

$$k^i = W^k a^i$$

v : information to be extracted

$$v^i = W^v a^i$$

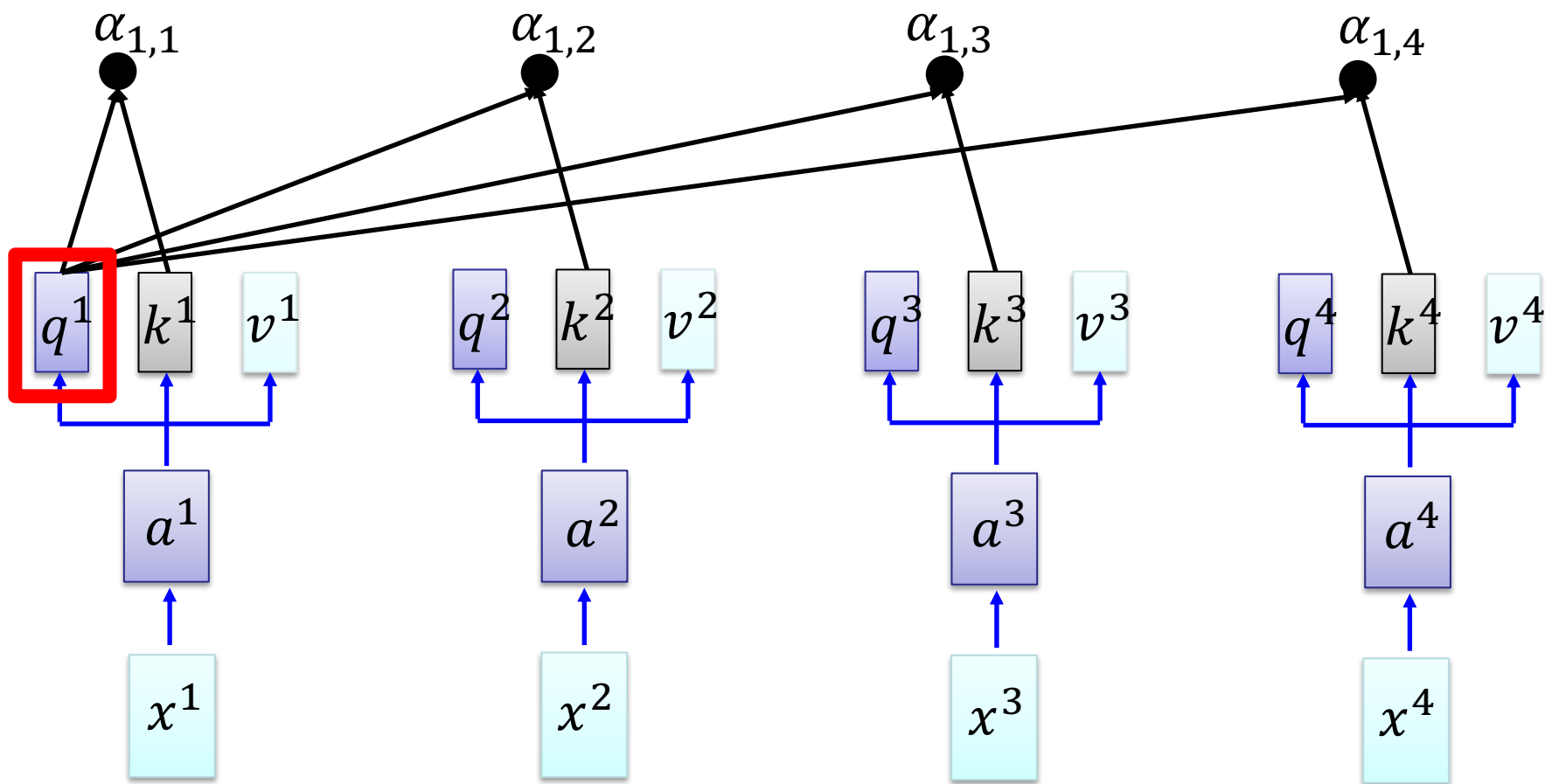


Self-attention

Take each query q , go to each key k , do attention

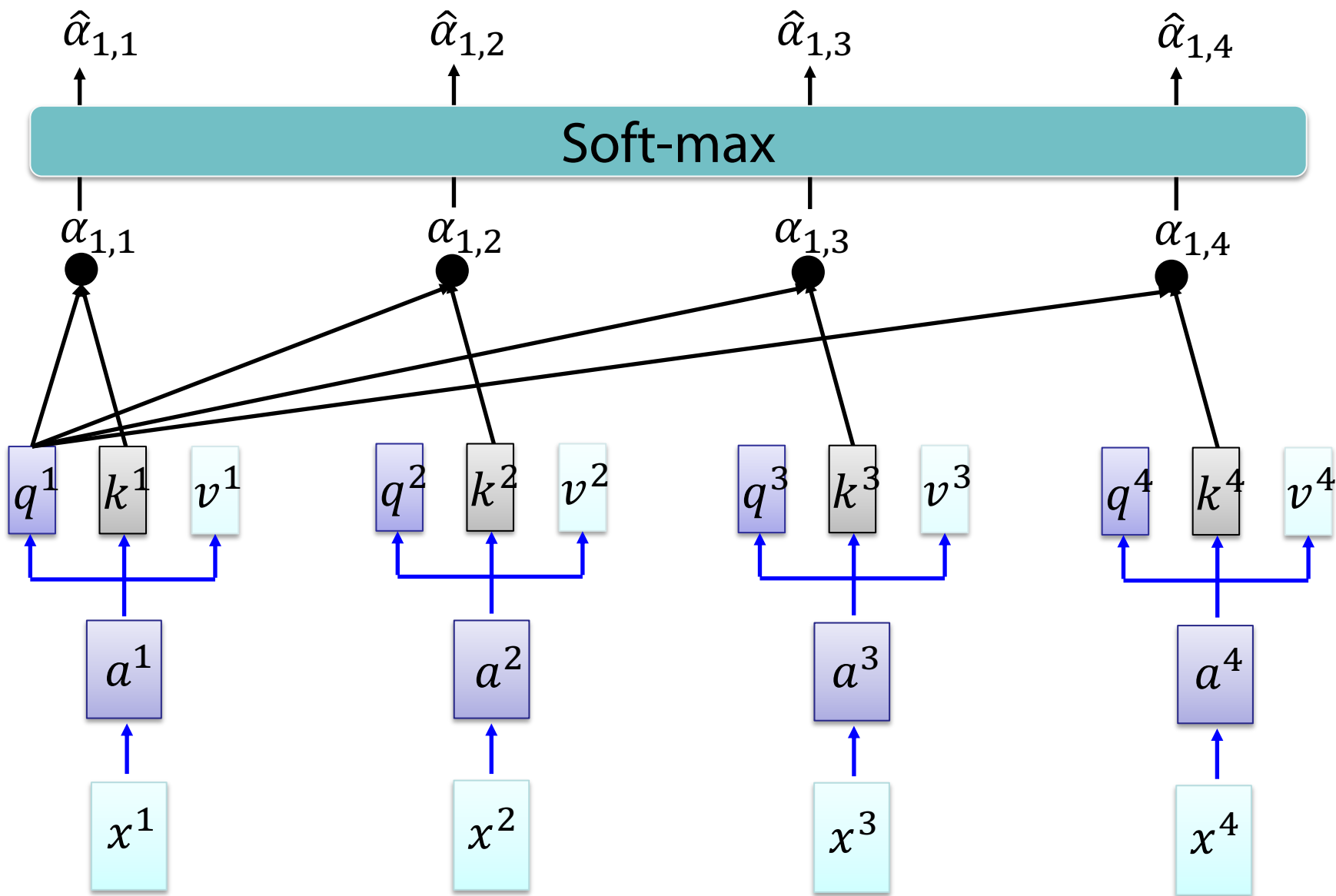
d is the dim of q and k

$$\text{Scaled Dot-Product Attention: } \alpha_{1,i} = \underbrace{q^1 \cdot k^i}_{\text{dot product}} / \sqrt{d}$$



Self-attention

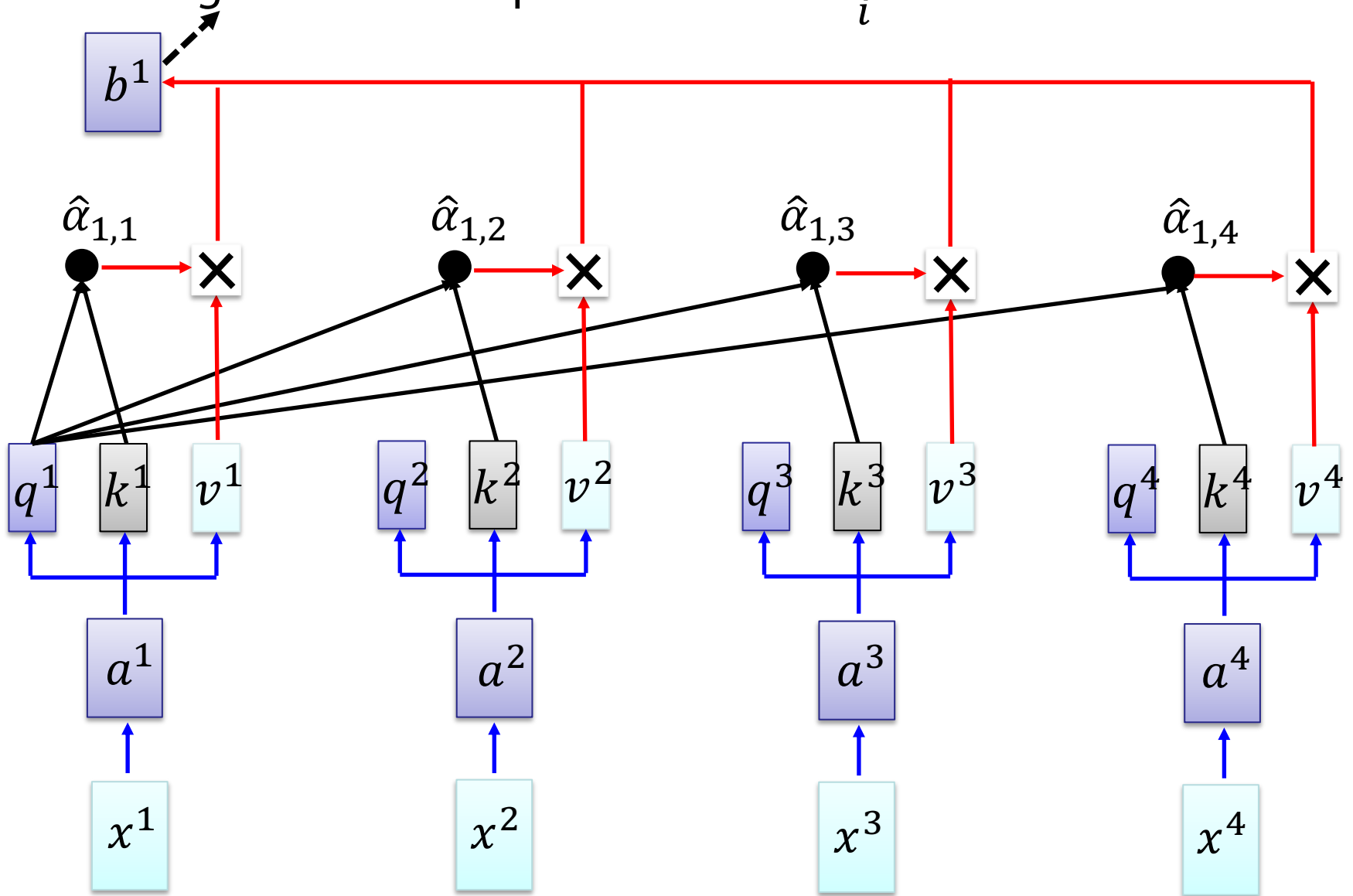
$$\hat{\alpha}_{1,i} = \exp(\alpha_{1,i}) / \sum_j \exp(\alpha_{1,j})$$



Self-attention

Considering the whole sequence

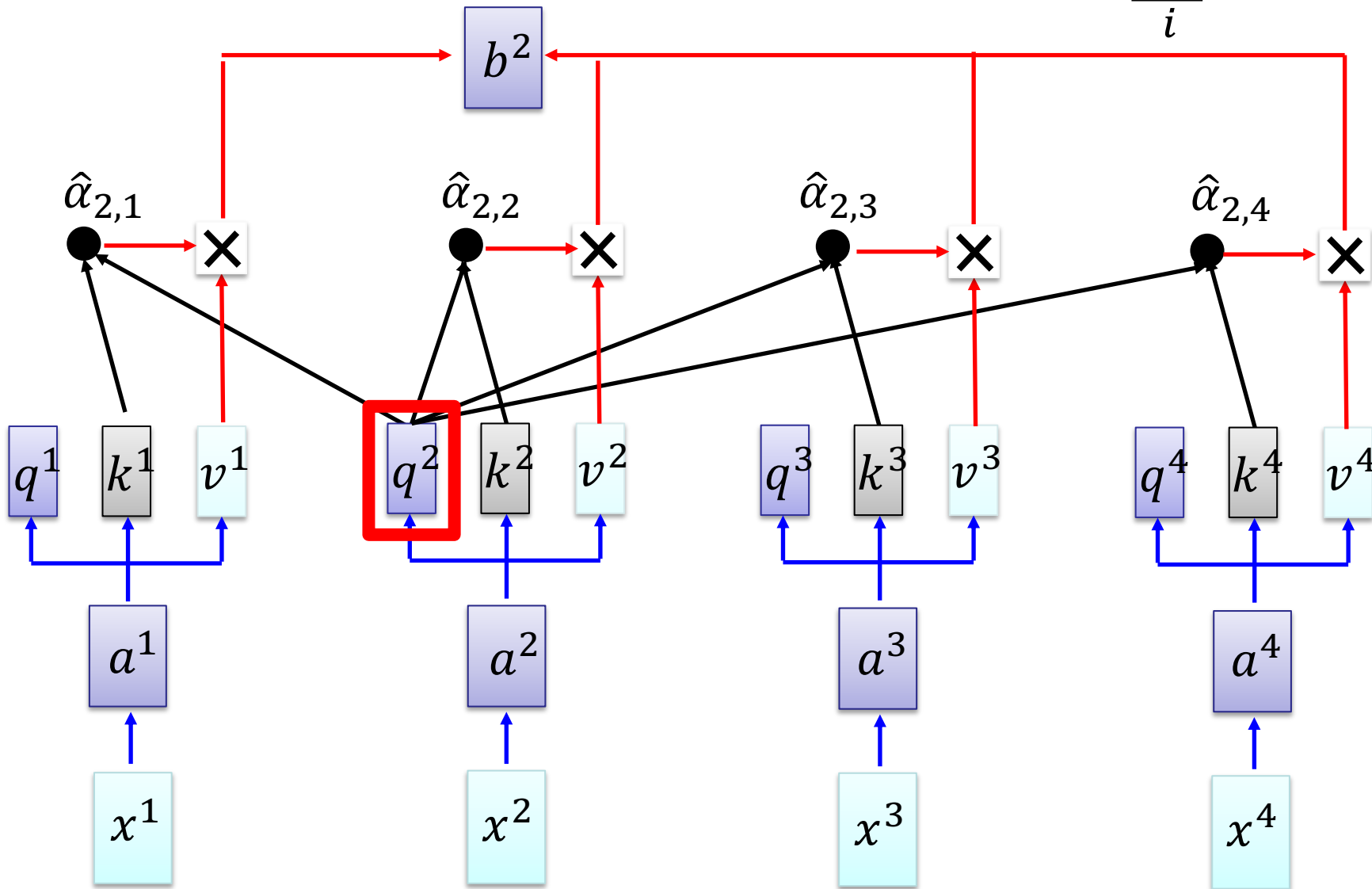
$$b^1 = \sum_i \hat{\alpha}_{1,i} v^i$$



Self-attention

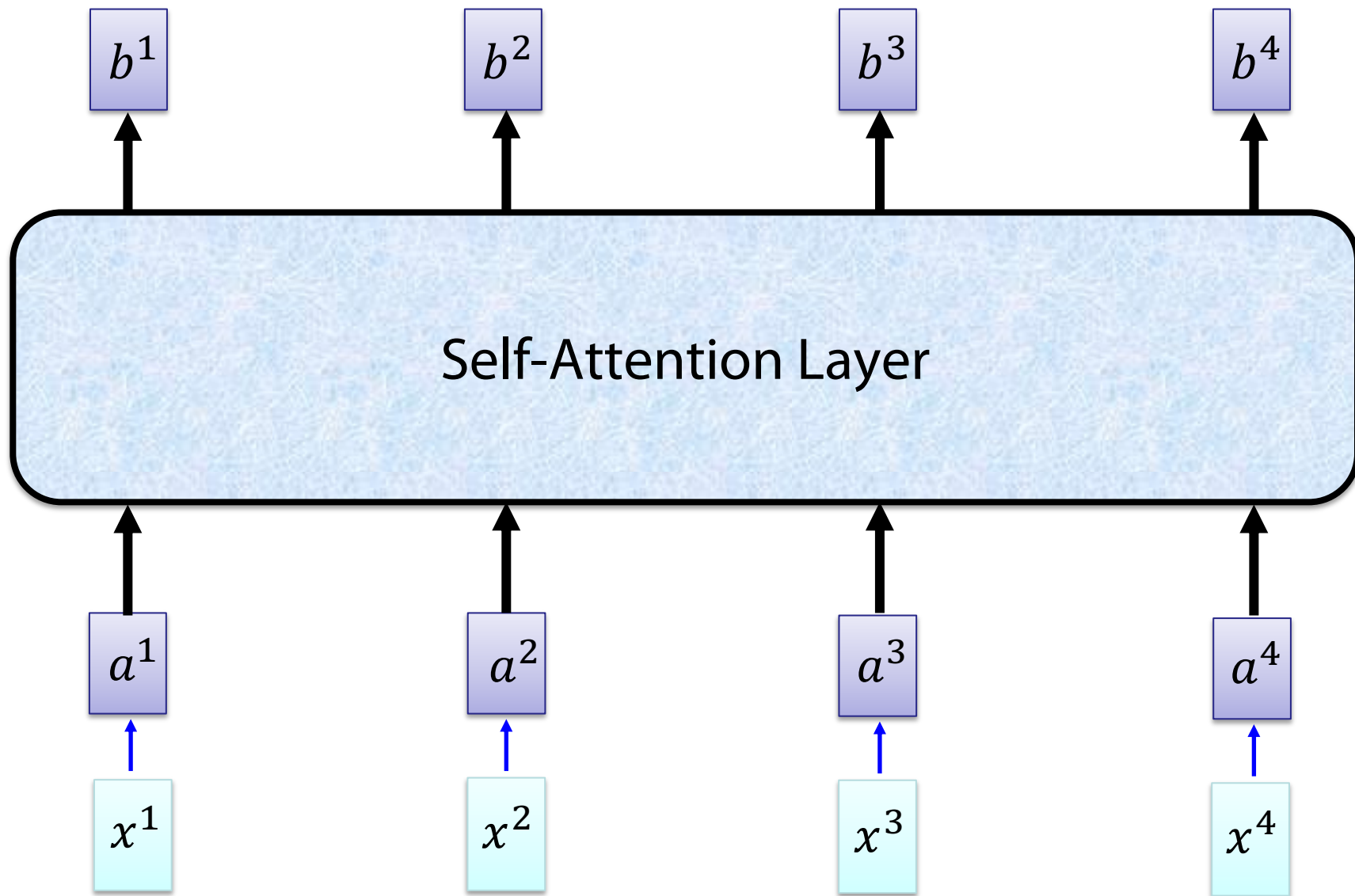
Take each query q , go to each key k , do attention

$$b^2 = \sum_i \hat{\alpha}_{2,i} v^i$$



Self-attention

b^1, b^2, b^3, b^4 can be computed in parallel



Self-attention

$$\begin{matrix} q^1 & q^2 & q^3 & q^4 \\ \hline Q \end{matrix} = \begin{matrix} W^q & \\ \hline \end{matrix} \begin{matrix} a^1 & a^2 & a^3 & a^4 \\ \hline I \end{matrix}$$

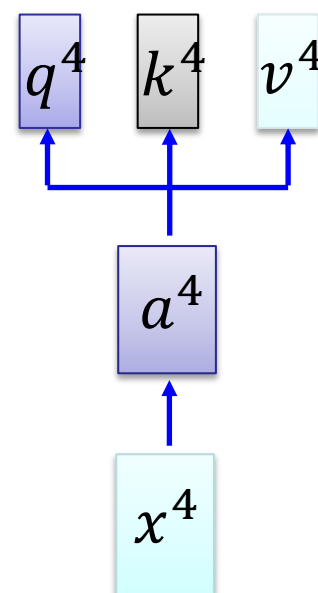
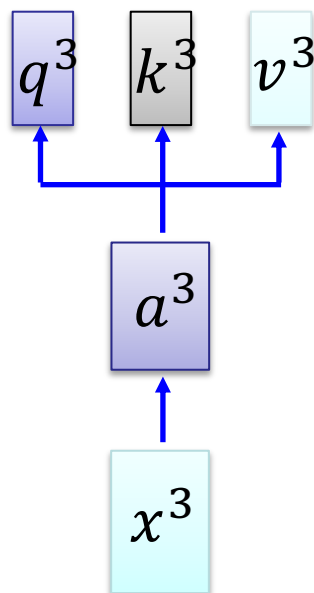
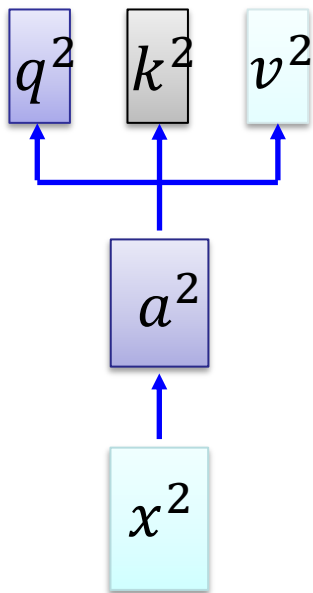
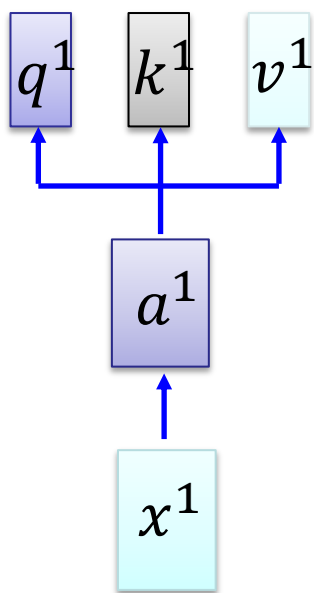
$$q^i = W^q a^i$$

$$\begin{matrix} k^1 & k^2 & k^3 & k^4 \\ \hline K \end{matrix} = \begin{matrix} W^k & \\ \hline \end{matrix} \begin{matrix} a^1 & a^2 & a^3 & a^4 \\ \hline I \end{matrix}$$

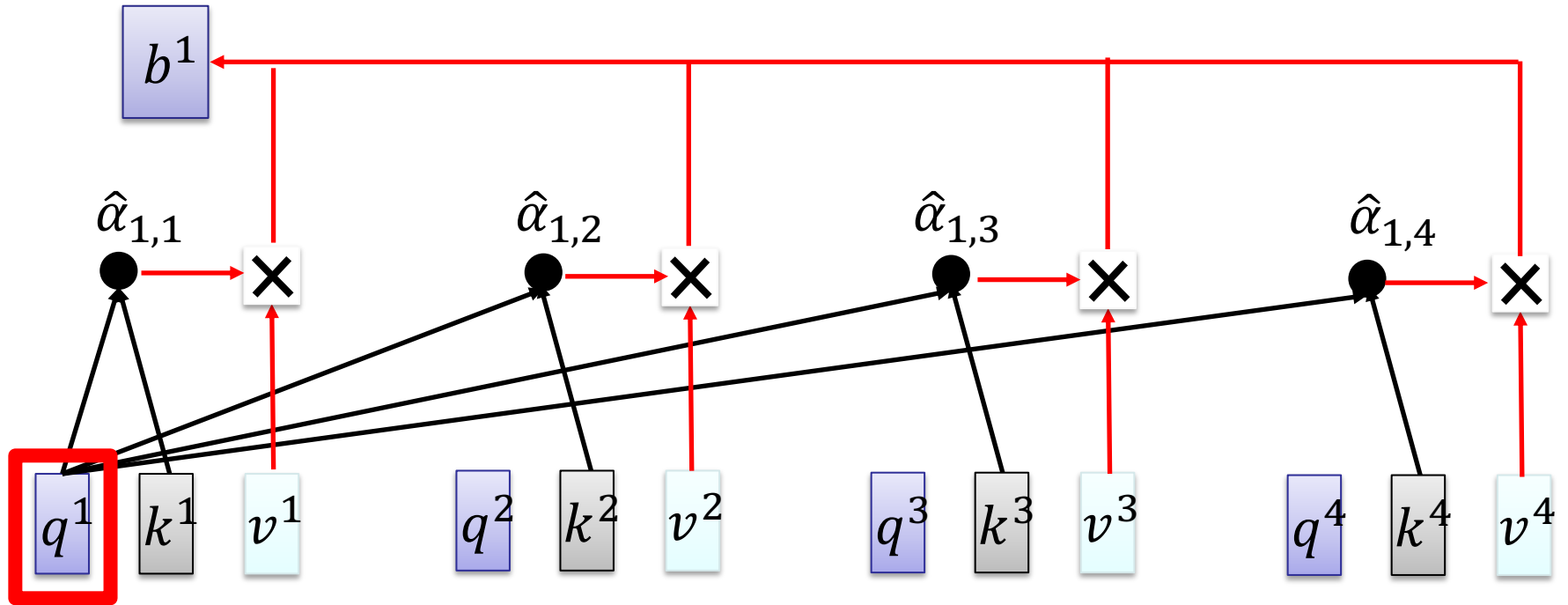
$$k^i = W^k a^i$$

$$\begin{matrix} v^1 & v^2 & v^3 & v^4 \\ \hline V \end{matrix} = \begin{matrix} W^v & \\ \hline \end{matrix} \begin{matrix} a^1 & a^2 & a^3 & a^4 \\ \hline I \end{matrix}$$

$$v^i = W^v a^i$$



Self-attention



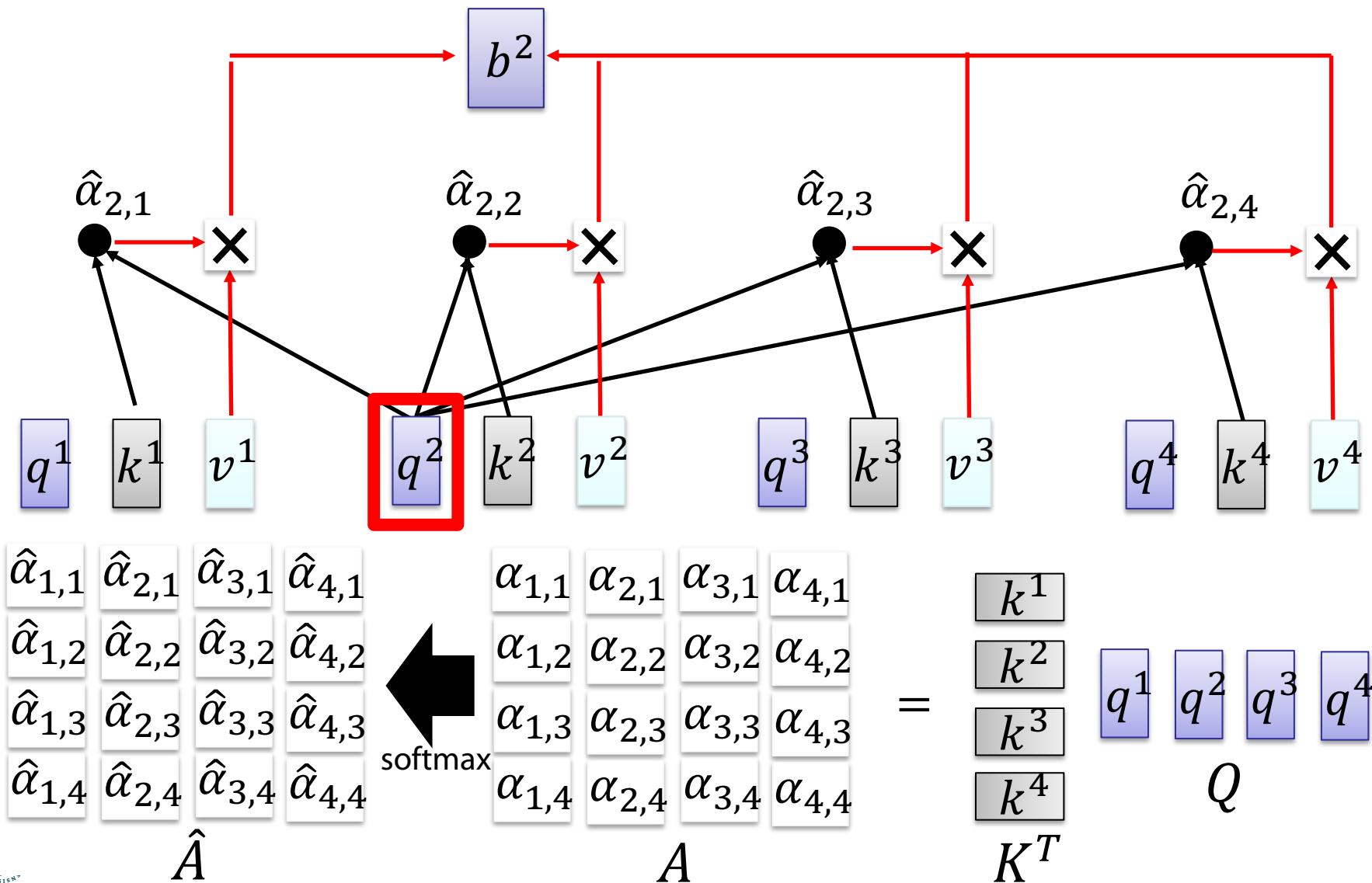
$$\alpha_{1,1} = k^1 q^1 \quad \alpha_{1,2} = k^2 q^1$$
$$\alpha_{1,3} = k^3 q^1 \quad \alpha_{1,4} = k^4 q^1$$

$$\begin{matrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{matrix} = \begin{matrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{matrix} q^1$$

(ignore \sqrt{d} for simplicity)

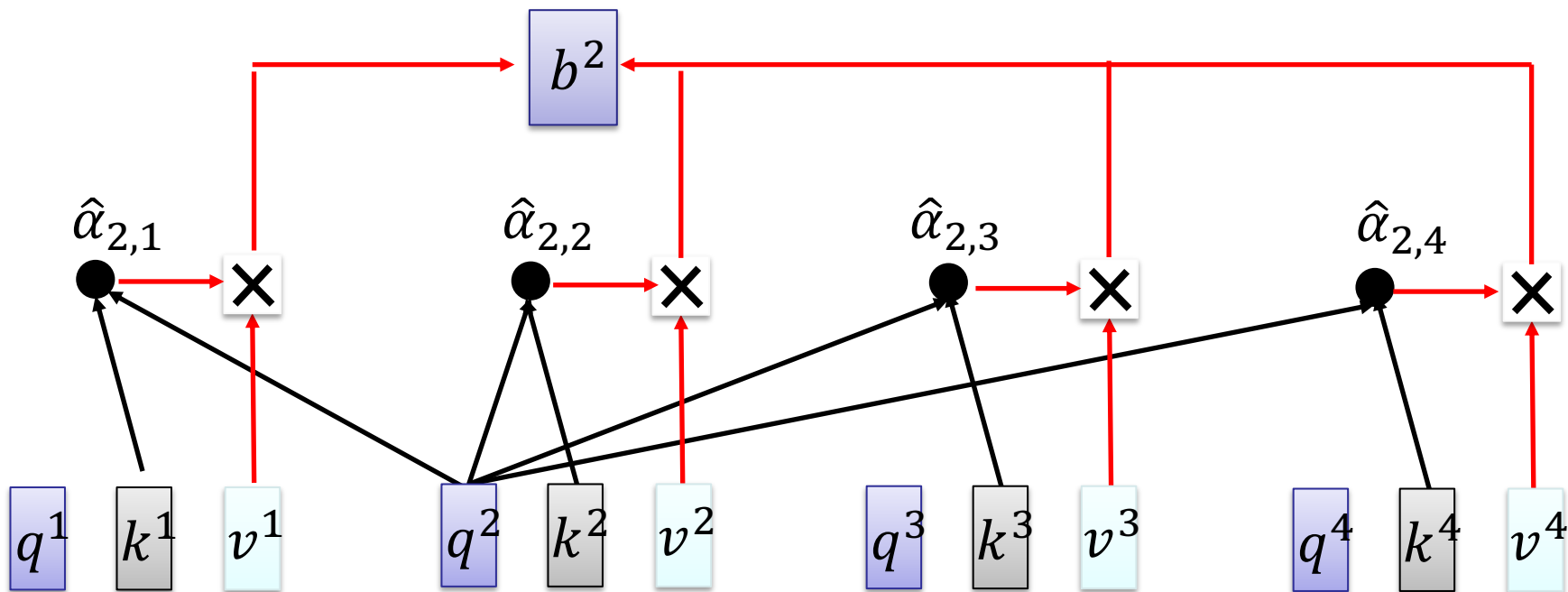
Self-attention

$$b^2 = \sum_i \hat{\alpha}_{2,i} v^i$$



Self-attention

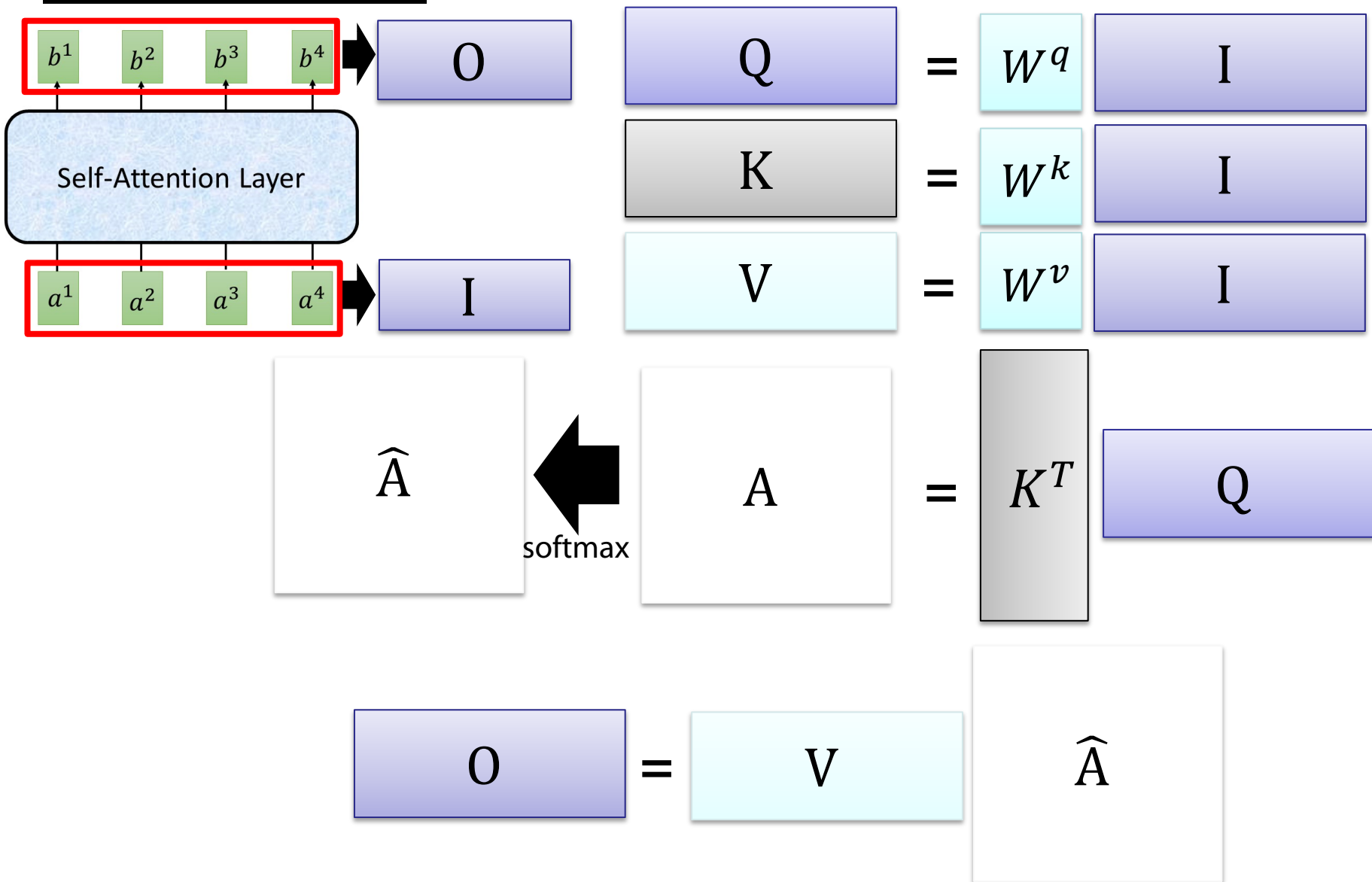
$$b^2 = \sum_i \hat{\alpha}_{2,i} v^i$$



$$\begin{matrix} b^1 & b^2 & b^3 & b^4 \\ \hline \end{matrix} \quad 0 = \begin{matrix} v^1 & v^2 & v^3 & v^4 \\ \hline \end{matrix} \quad \hat{A}$$

$\hat{\alpha}_{1,1}$	$\hat{\alpha}_{2,1}$	$\hat{\alpha}_{3,1}$	$\hat{\alpha}_{4,1}$
$\hat{\alpha}_{1,2}$	$\hat{\alpha}_{2,2}$	$\hat{\alpha}_{3,2}$	$\hat{\alpha}_{4,2}$
$\hat{\alpha}_{1,3}$	$\hat{\alpha}_{2,3}$	$\hat{\alpha}_{3,3}$	$\hat{\alpha}_{4,3}$
$\hat{\alpha}_{1,4}$	$\hat{\alpha}_{2,4}$	$\hat{\alpha}_{3,4}$	$\hat{\alpha}_{4,4}$

Self-attention



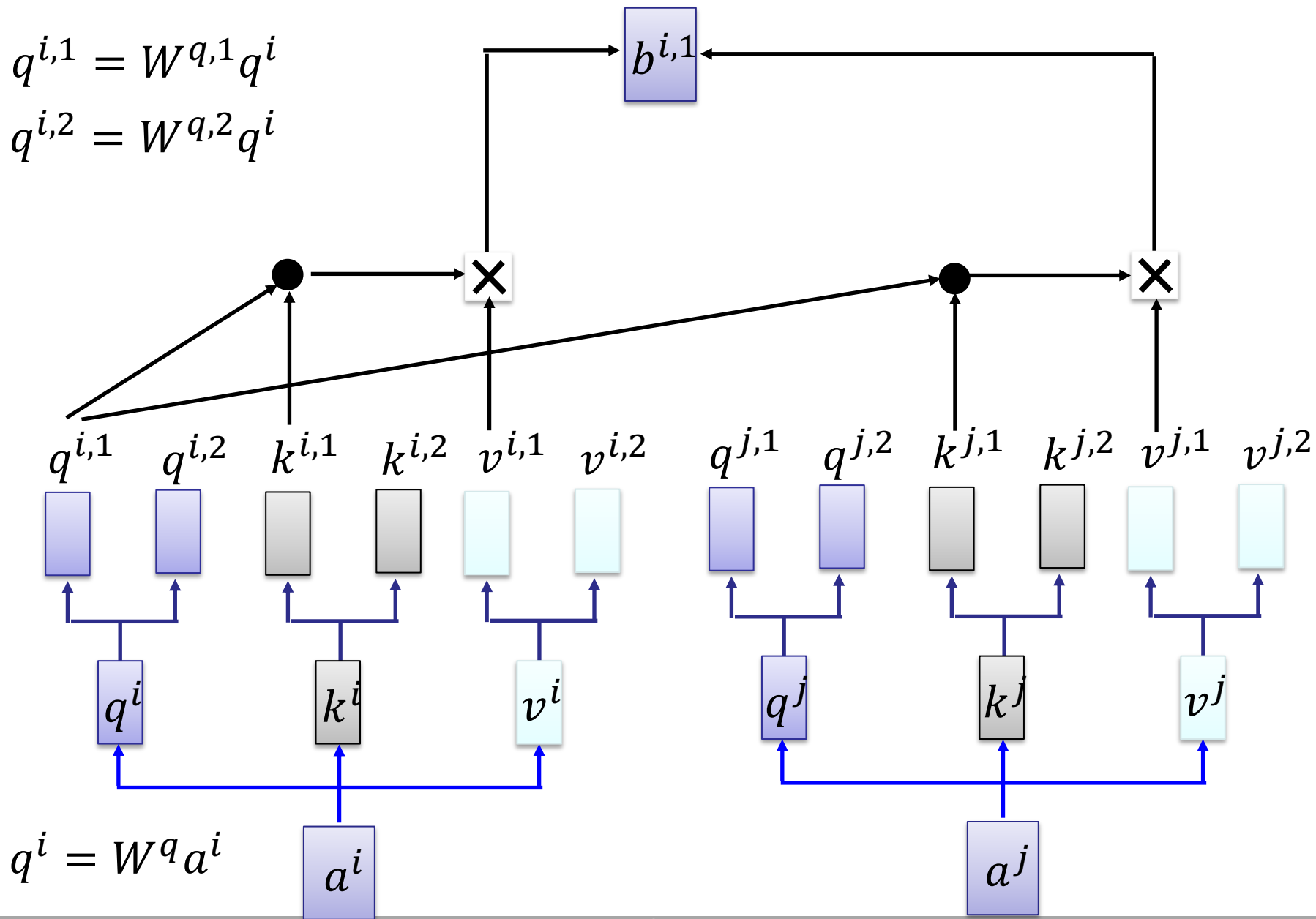
Can be optimized with GPUs

Multi-head Self-attention

(2 heads as example)

$$q^{i,1} = W^{q,1} q^i$$

$$q^{i,2} = W^{q,2} q^i$$

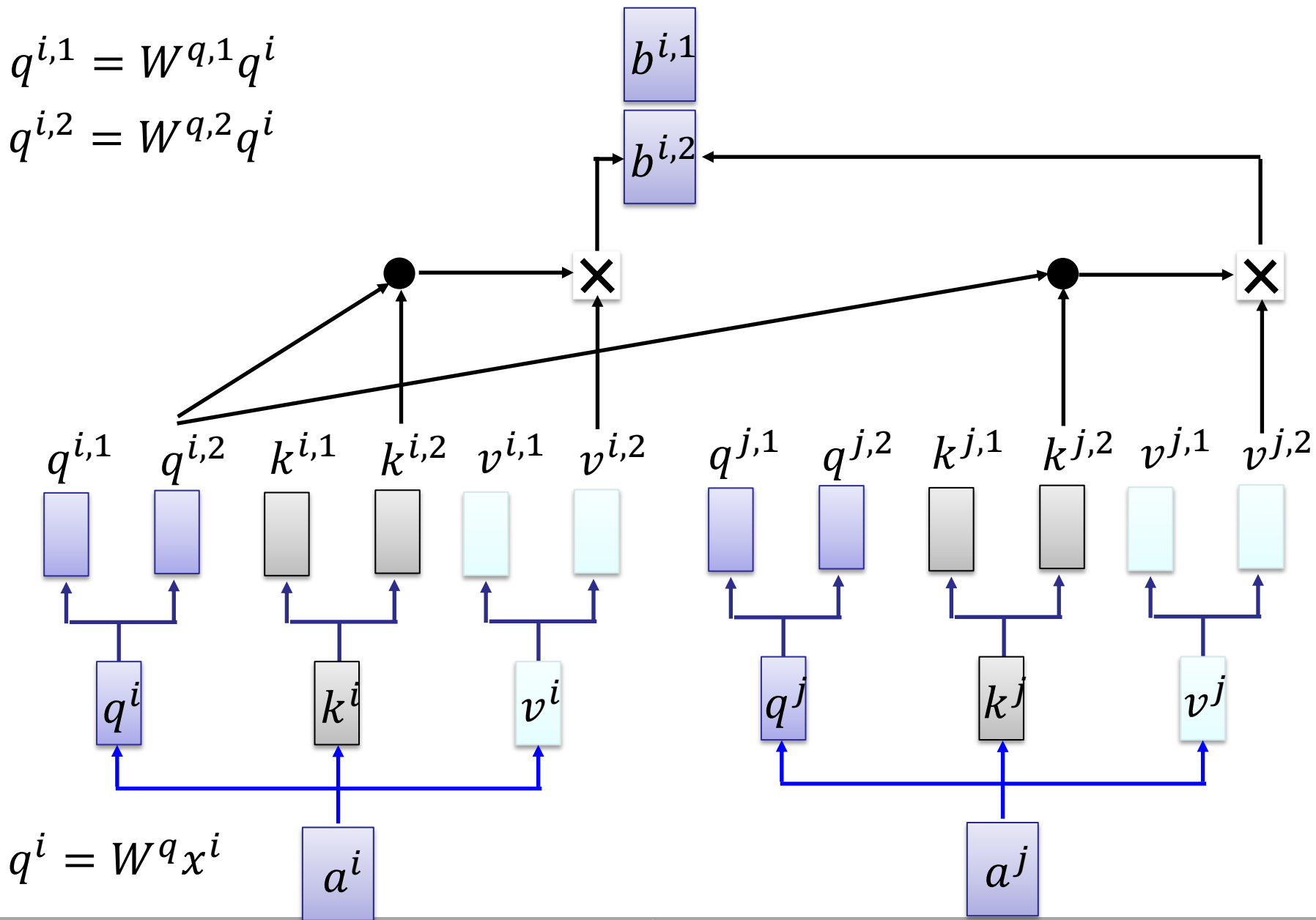


Multi-head Self-attention

(2 heads as example)

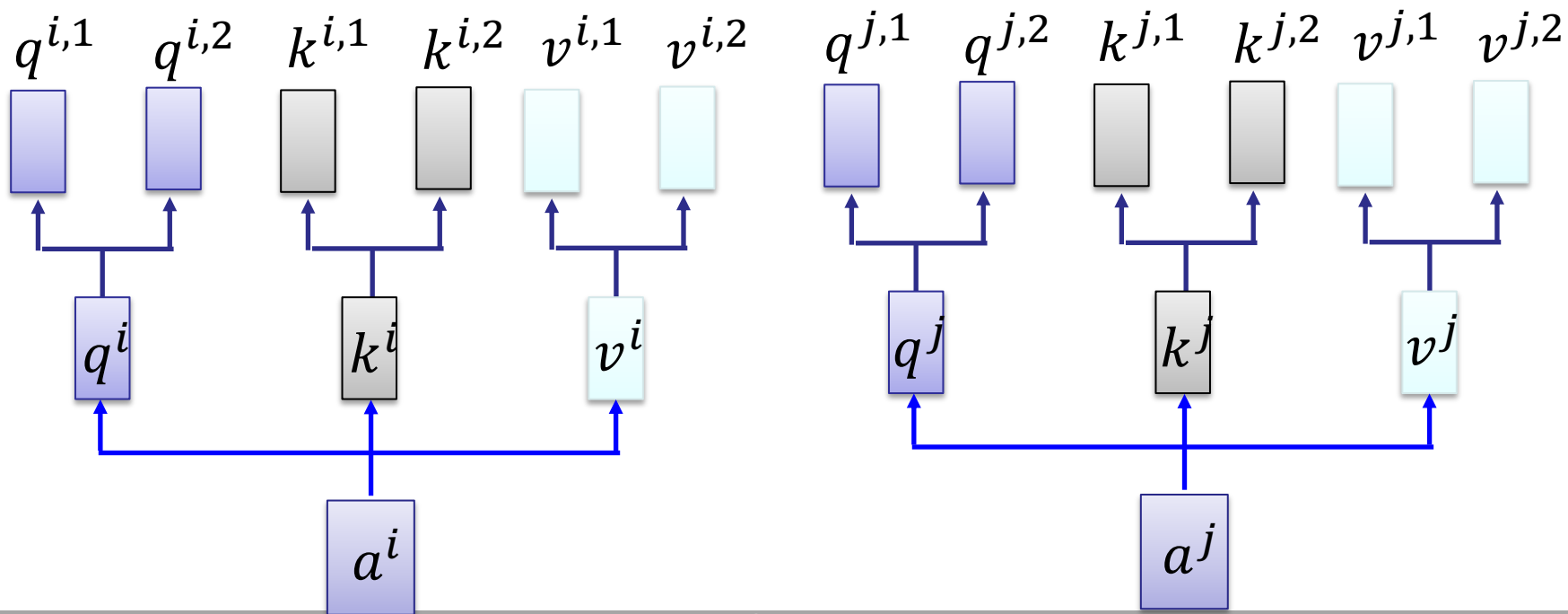
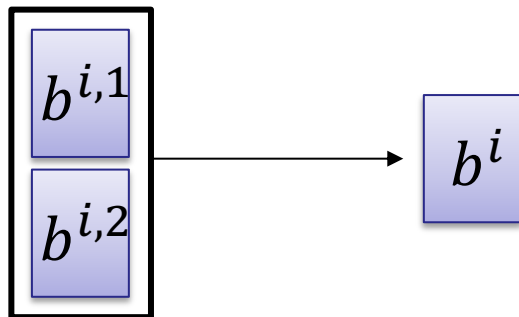
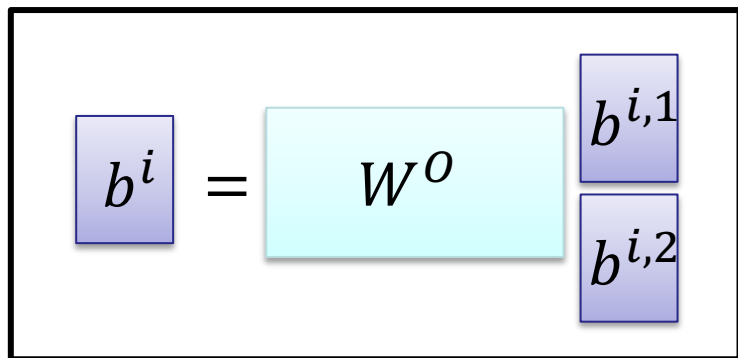
$$q^{i,1} = W^{q,1} q^i$$

$$q^{i,2} = W^{q,2} q^i$$



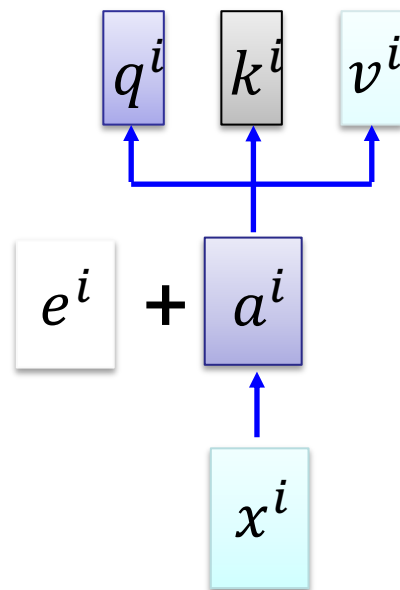
Multi-head Self-attention

(2 heads as example)

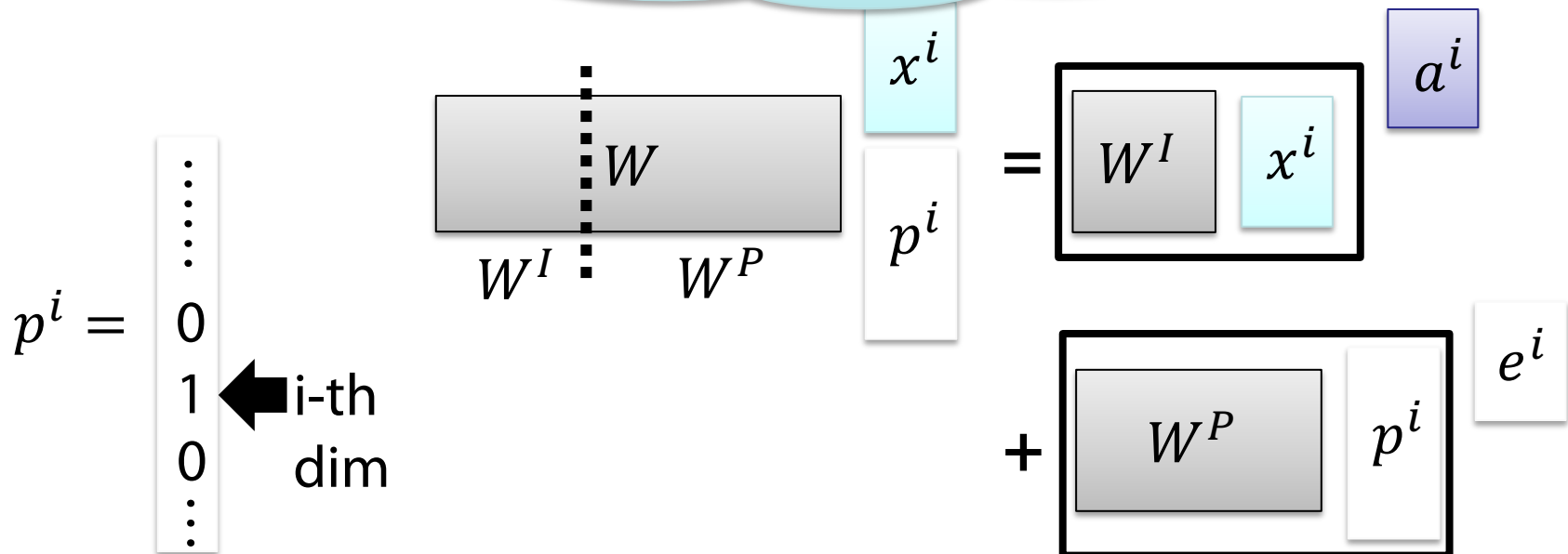


Positional Encoding

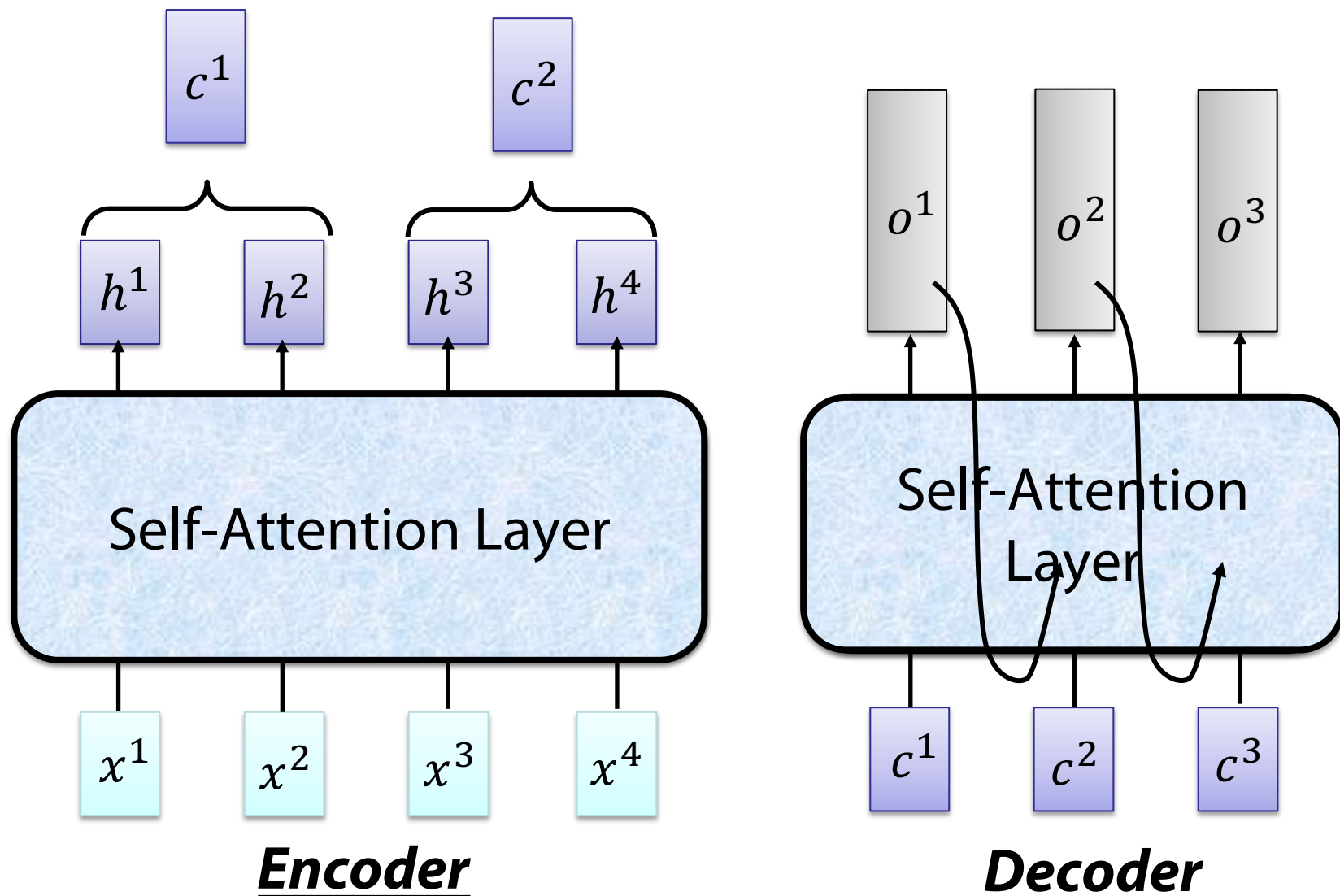
- No position information in self-attention.
- Each position has a unique positional vector e^i (not learned from data)
- Idea: Append each x^i is with a one-hot vector p^i



More clever solution used in the original paper



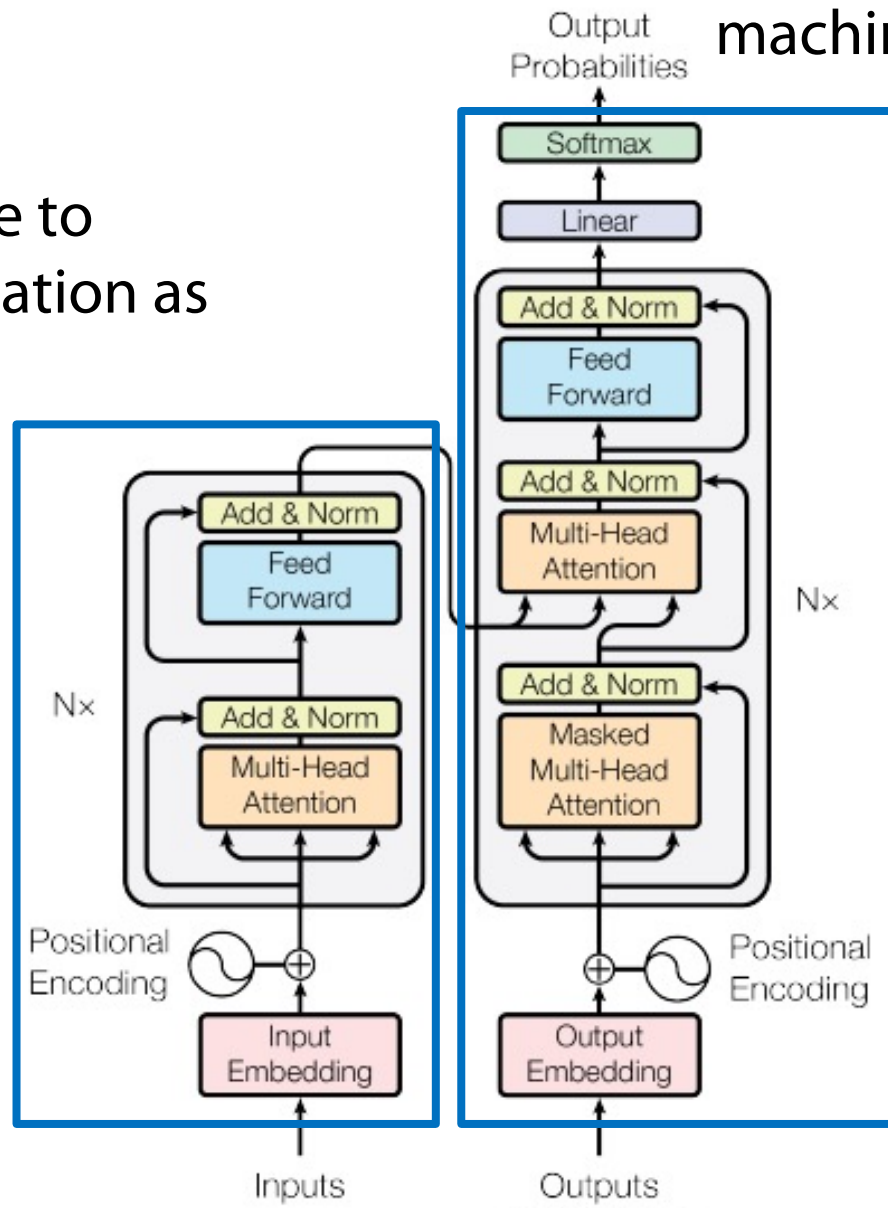
Seq2seq with Attention



Transformer

Using Chinese to English translation as example

Encoder



Decoder

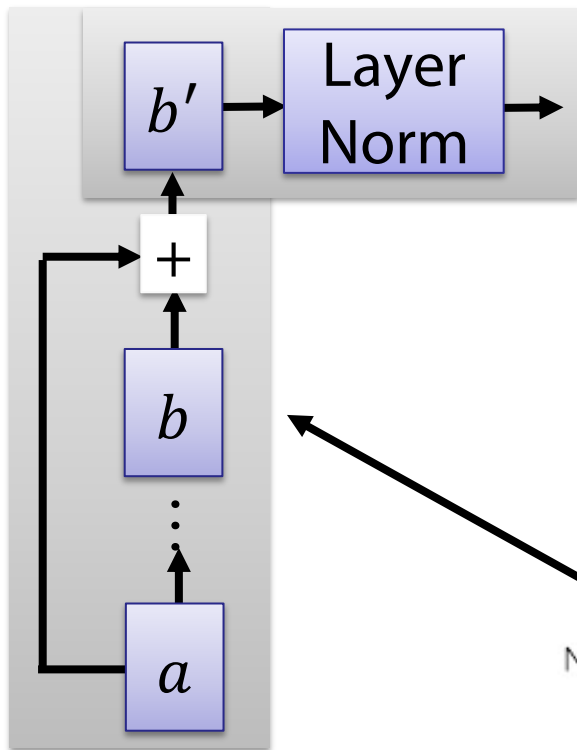
機器學習

Outputs (shifted right)

<BOS>

machine

Transformer

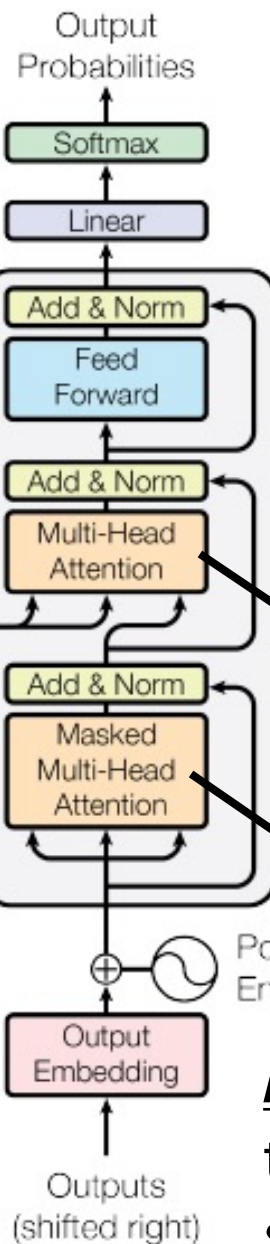
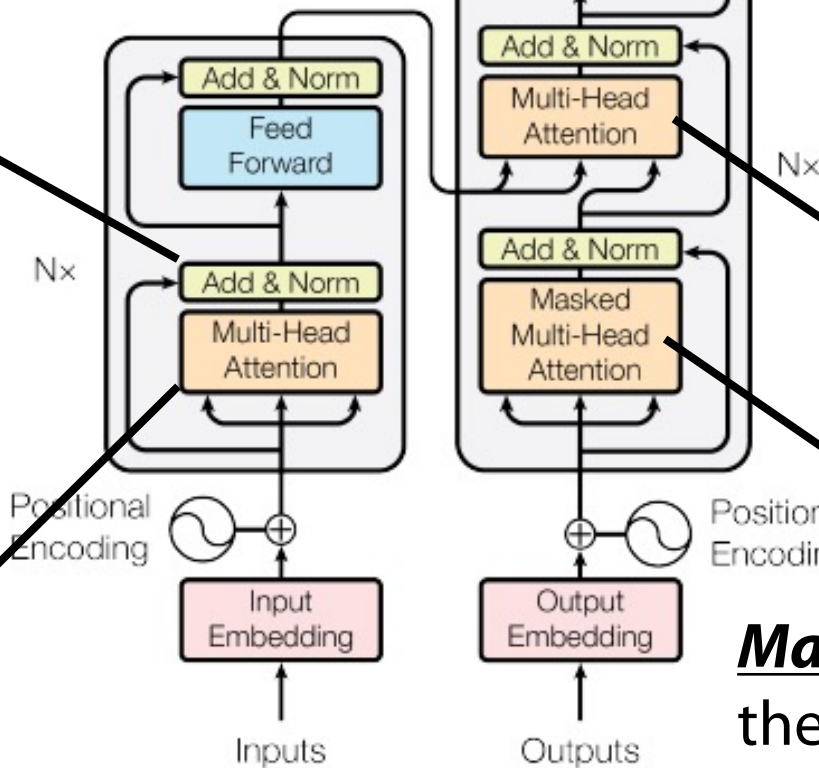
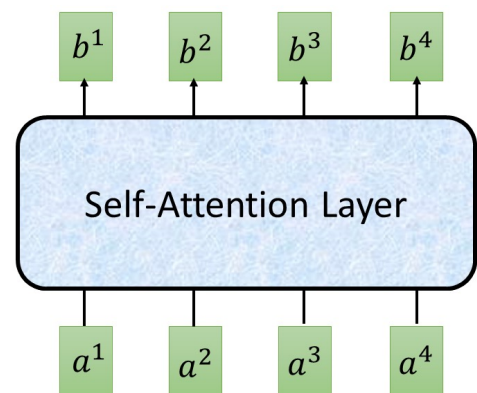


Layer Norm:

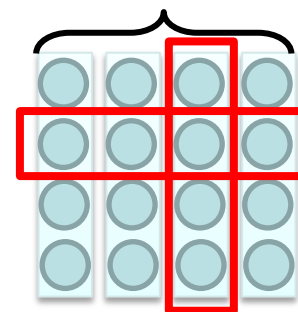
<https://arxiv.org/abs/1607.06450>

Batch Norm:

<https://www.youtube.com/watch?v=BZh1ltr5Rkg>



Batch Size



$\mu = 0,$
 $\sigma = 1$
Batch

$\mu = 0, \sigma = 1$
Layer

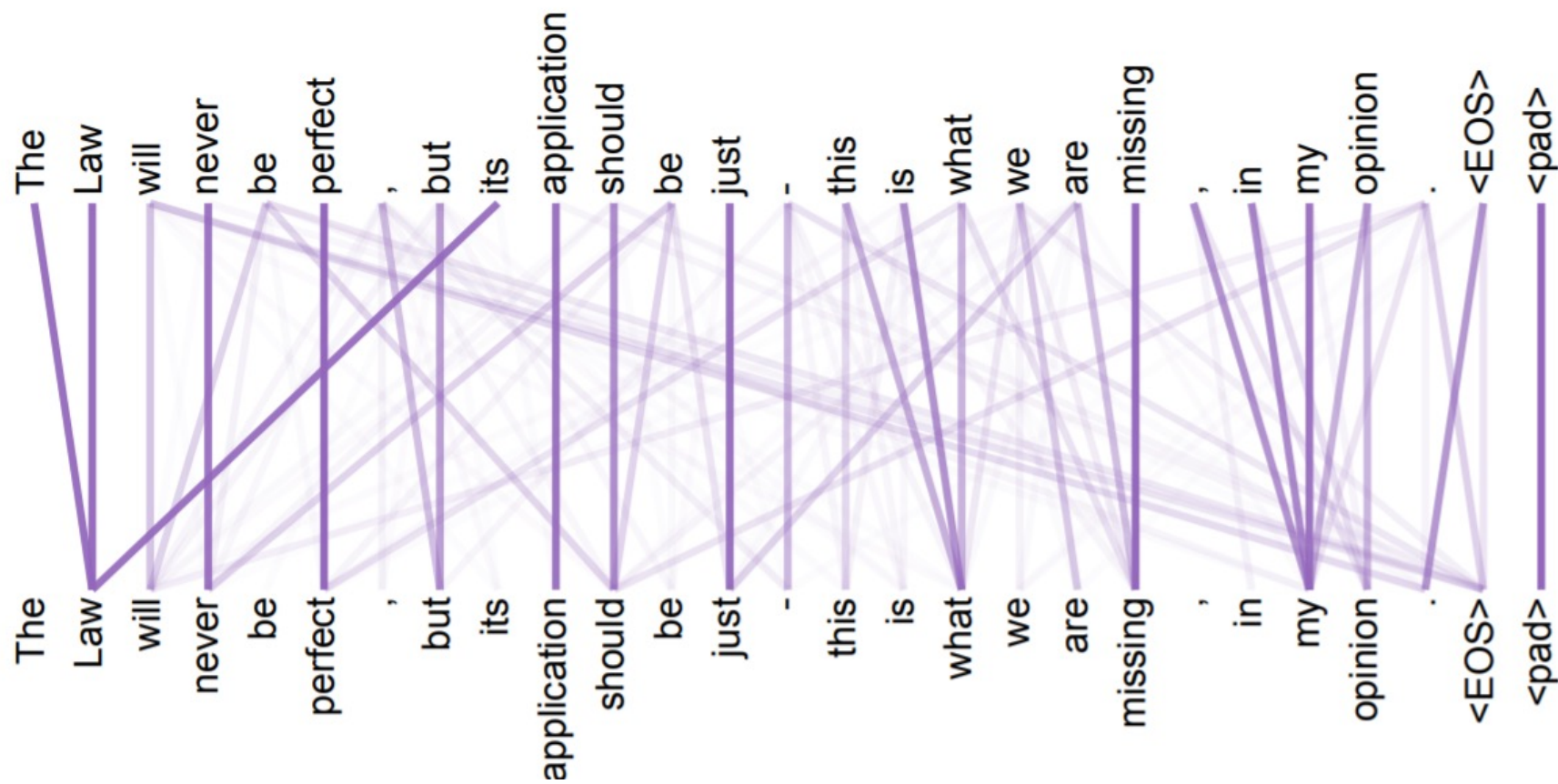
attend on the
input sequence

Masked: attend on
the generated
sequence [MASK]

Masked Multihead Attention

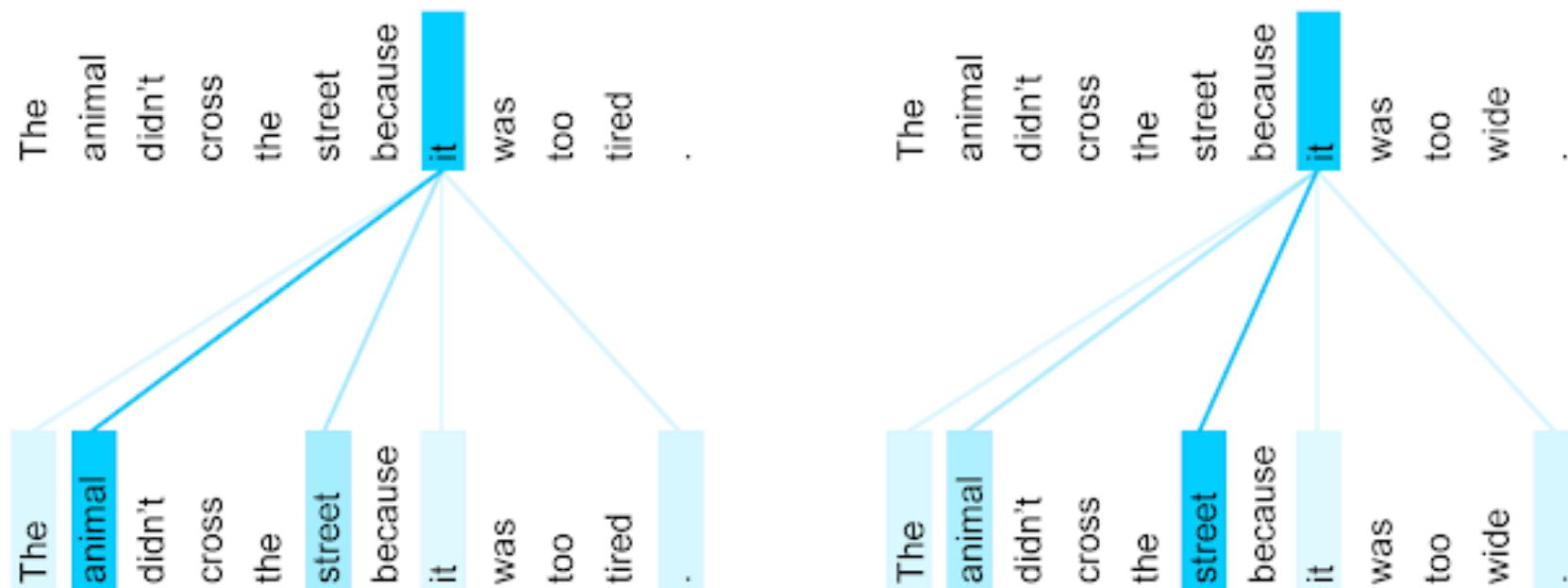
- Decoder should work in parallel as well
- During training all output tokens are known
- Copy output #token times
- For each position use [MASK] token in copies
- Attention becomes possible during training also for decoding
- Train decoder such that [MASK] is replaced correctly while paying attention to the overall output training data

Attention Visualization



<https://arxiv.org/abs/1706.03762>

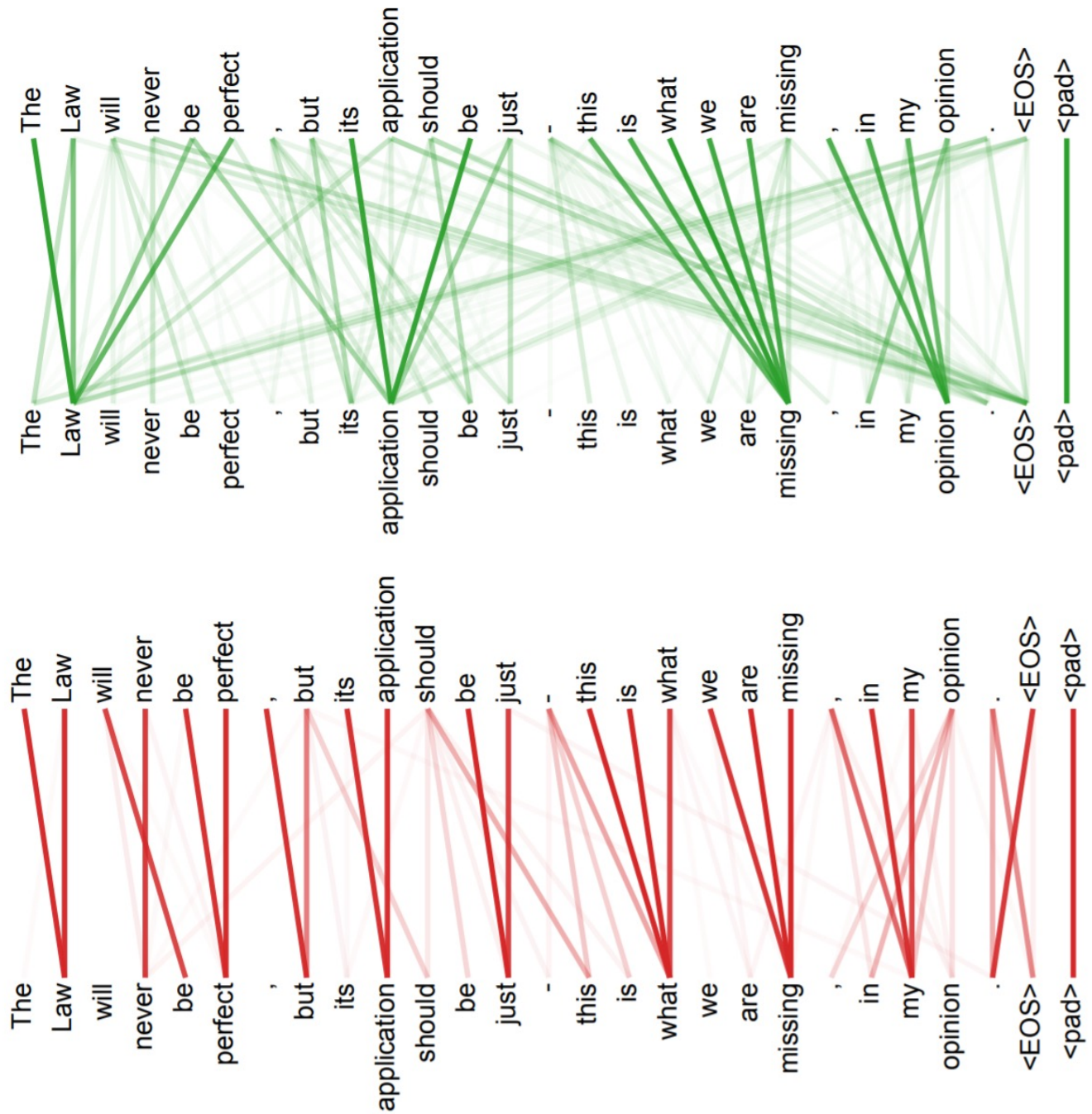
Attention Visualization



The encoder self-attention distribution for the word "it" from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads).

<https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html>

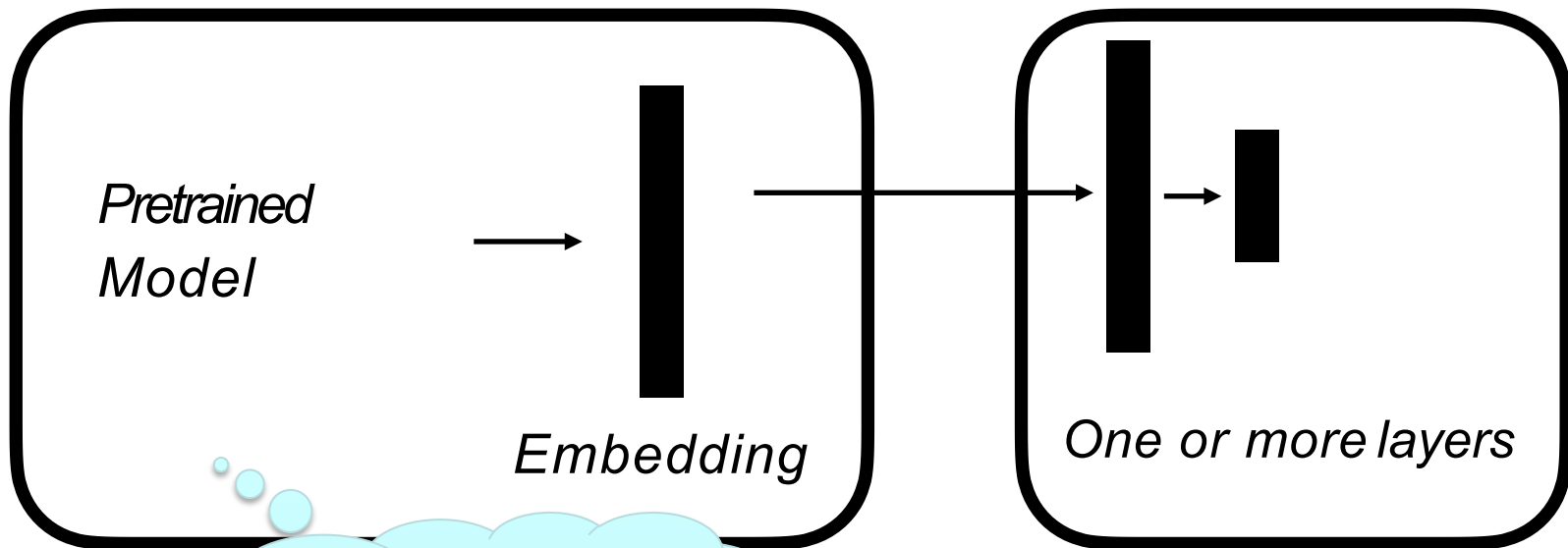
Multi-head Attention



Pre-Training & Fine Tuning

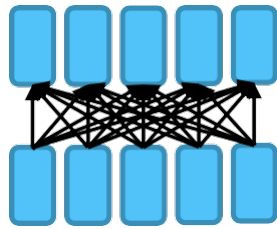
1) Download LM
pre-trained on large corpus
(in self-supervised fashion)

2) Feature-based training ("fine-tuning")
on target task
(supervised learning)



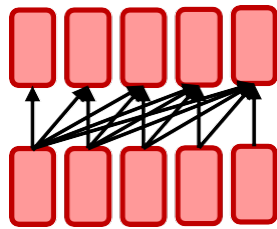
Usually frozen
after pre-training

Model Pre-Training



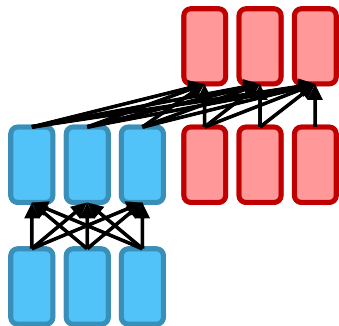
- Encoder

- Bidirectional context
- Examples: BERT and its variants



- Decoder

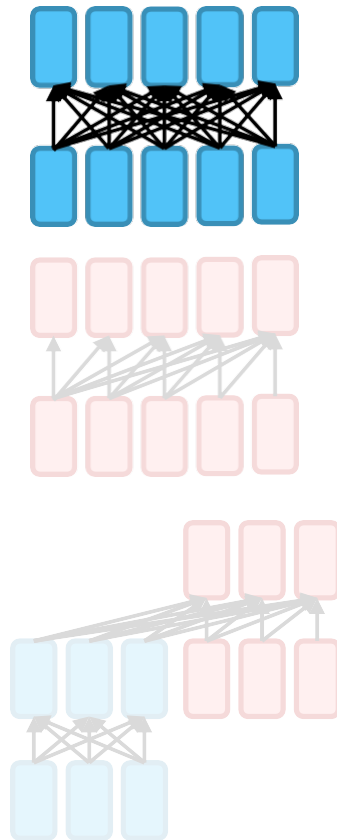
- Language modeling; better for generation
- Example: GPT-2, GPT-3, [LaMDA](#)



- Encoder-Decoder

- Sequence-to-sequence model
- Examples: Transformer, BART, T5

Model Pre-Training

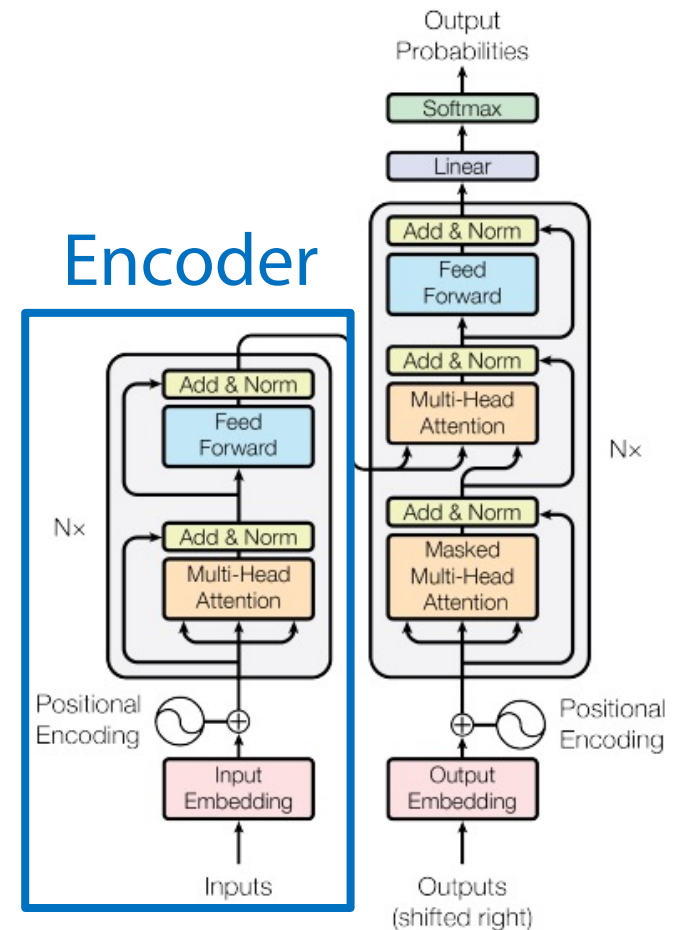
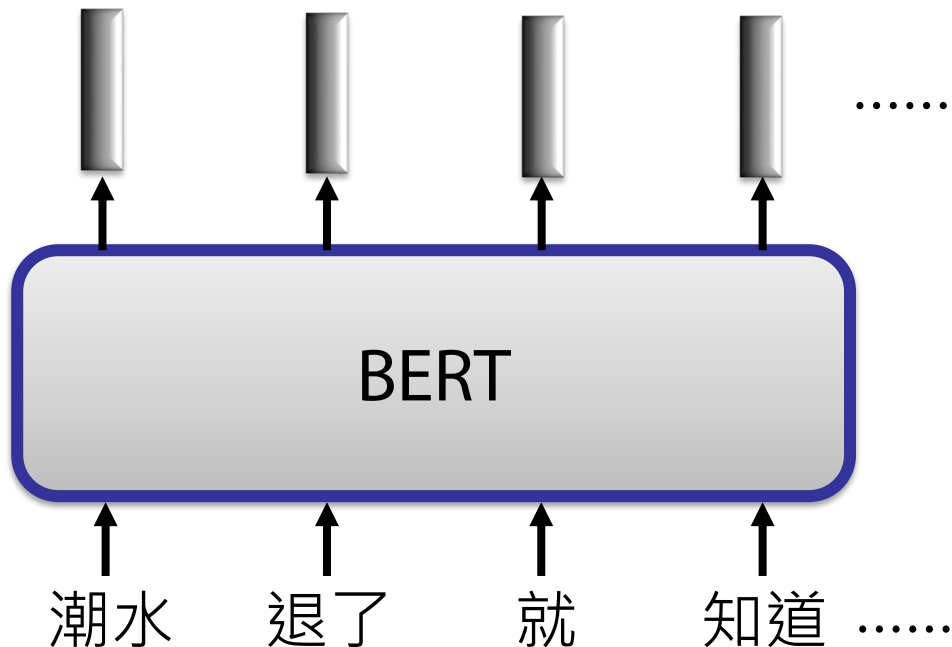


- **Encoder**
 - Bidirectional context
 - Examples: BERT and its variants
- **Decoder**
 - Language modeling; better for generation
 - Example: GPT-2, GPT-3, [LaMDA](#)
- **Encoder-Decoder**
 - Sequence-to-sequence model
 - Examples: Transformer, BART, T5

Bidirectional Encoder Representations from Transformers (BERT)

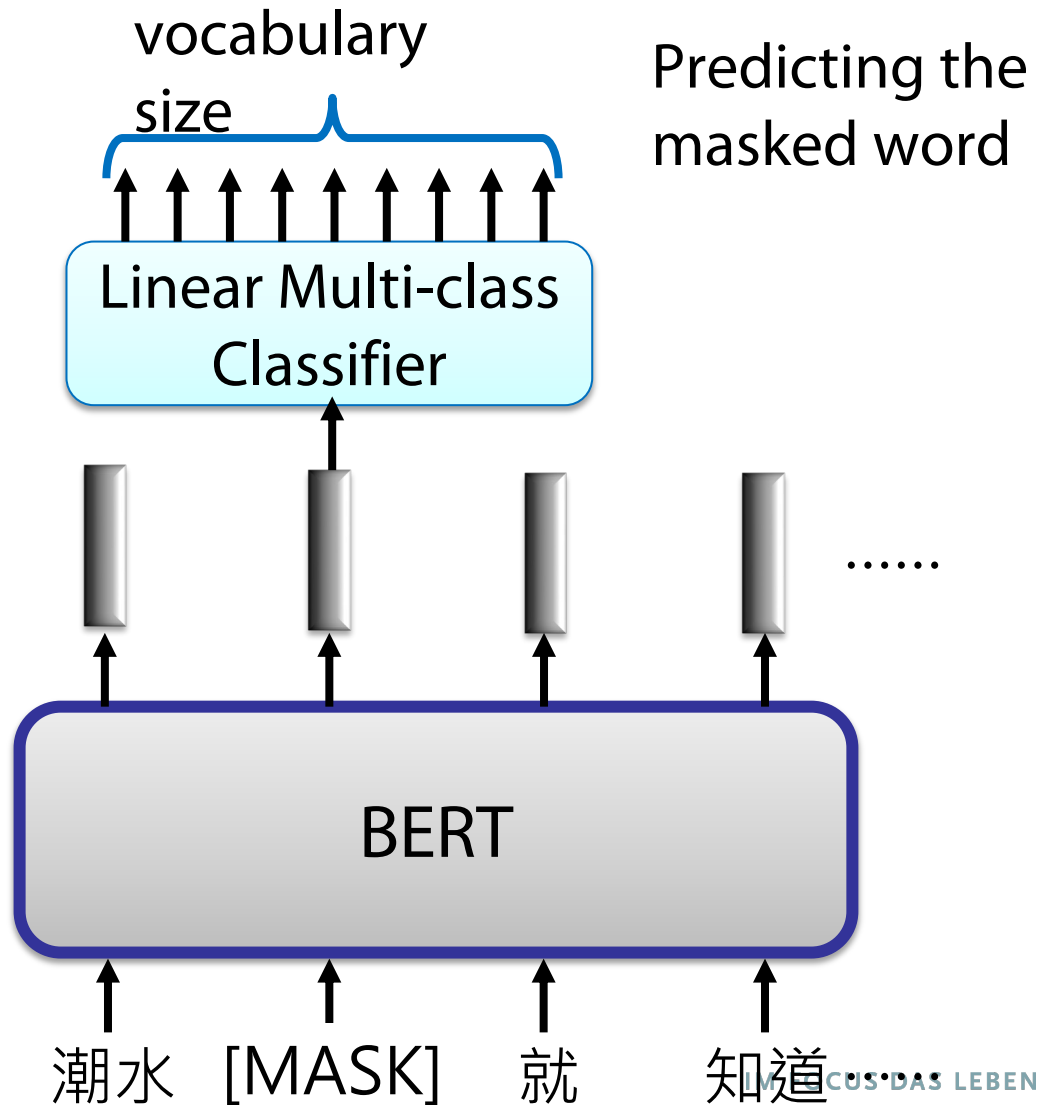
- BERT = Encoder of Transformer

Learned from a large amount of text without annotation

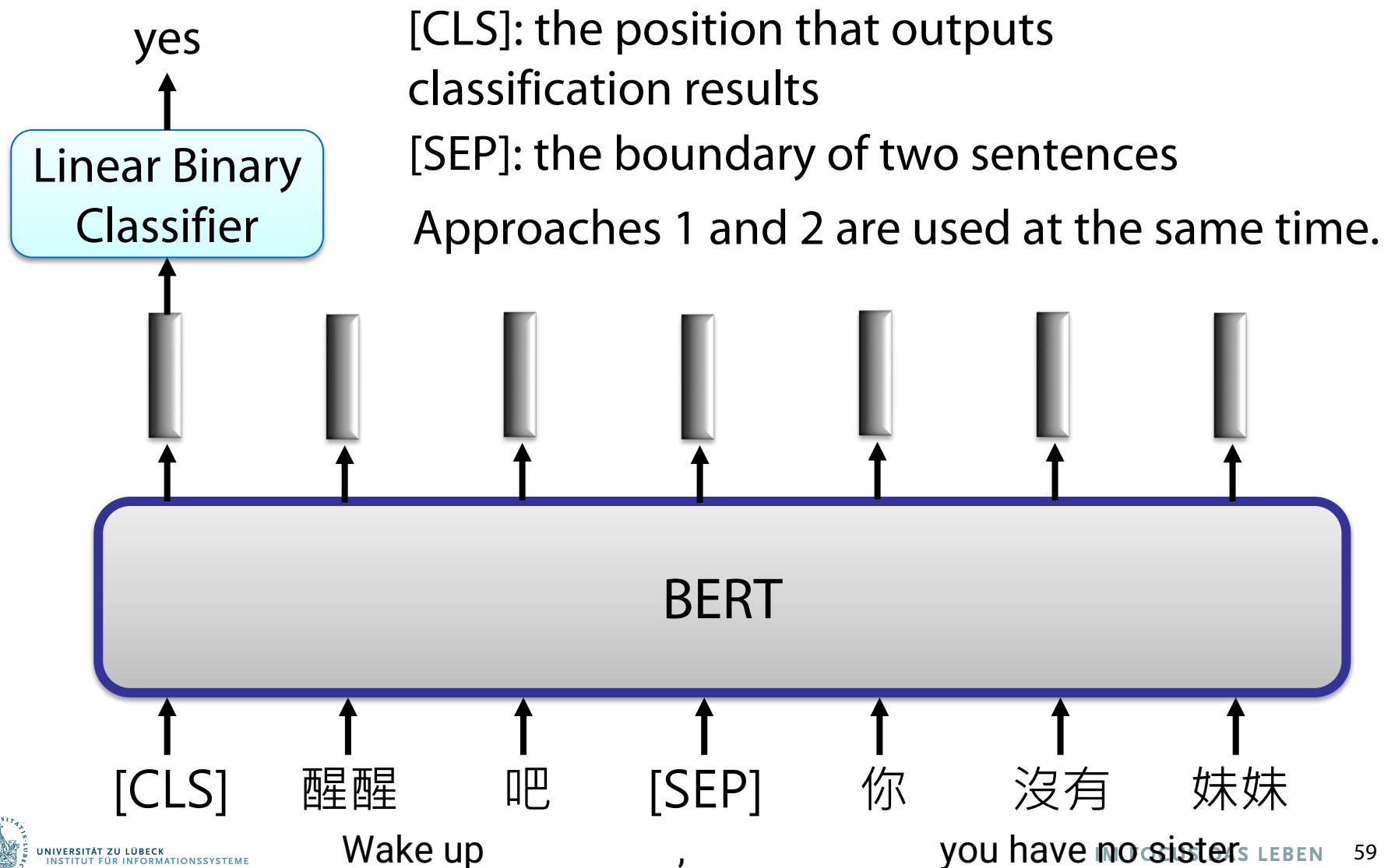


Training of BERT

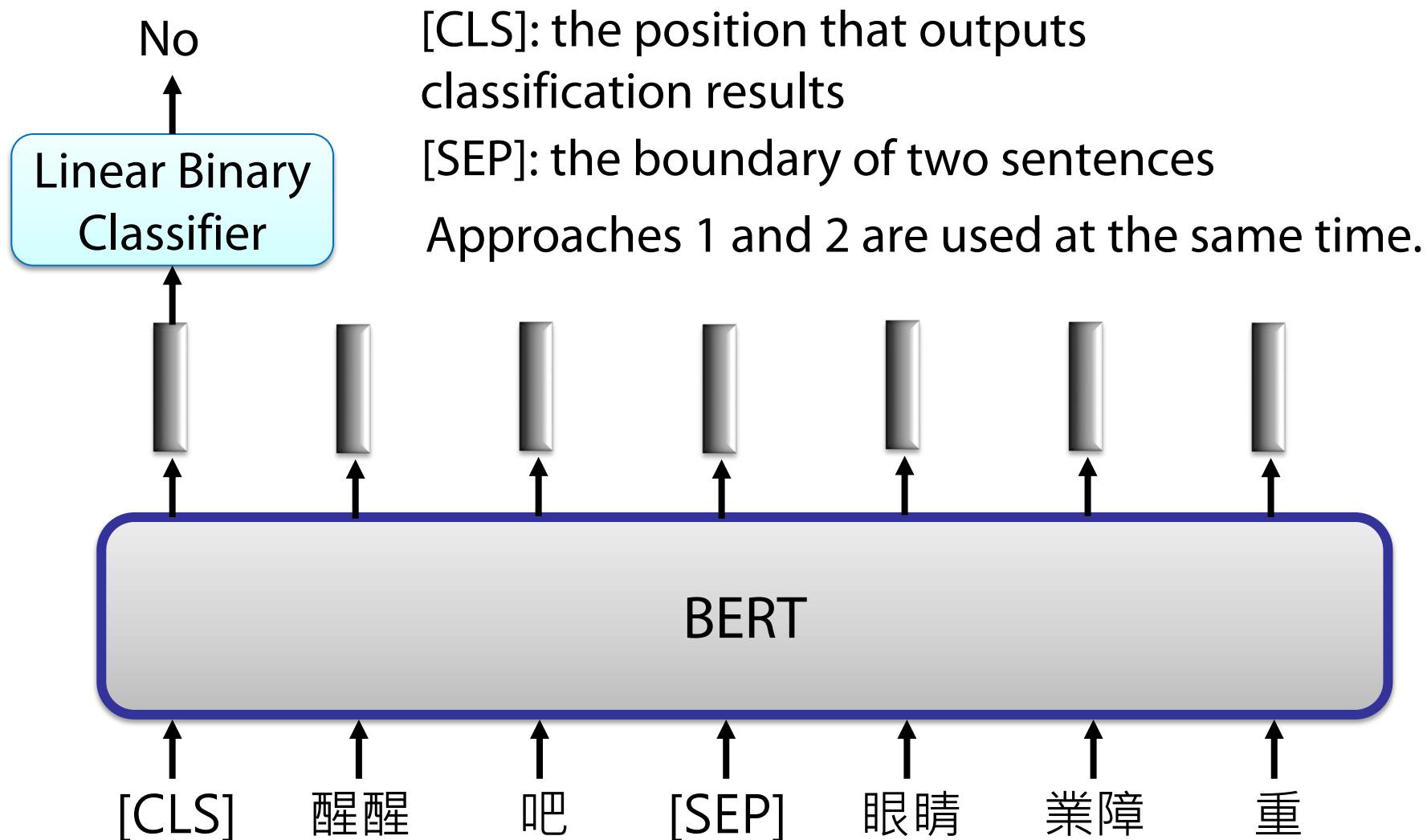
- Approach 1:
Masked LM



Training of BERT – Approach 2: Next Sentence Prediction



Training of BERT – Approach 2: Next Sentence Prediction

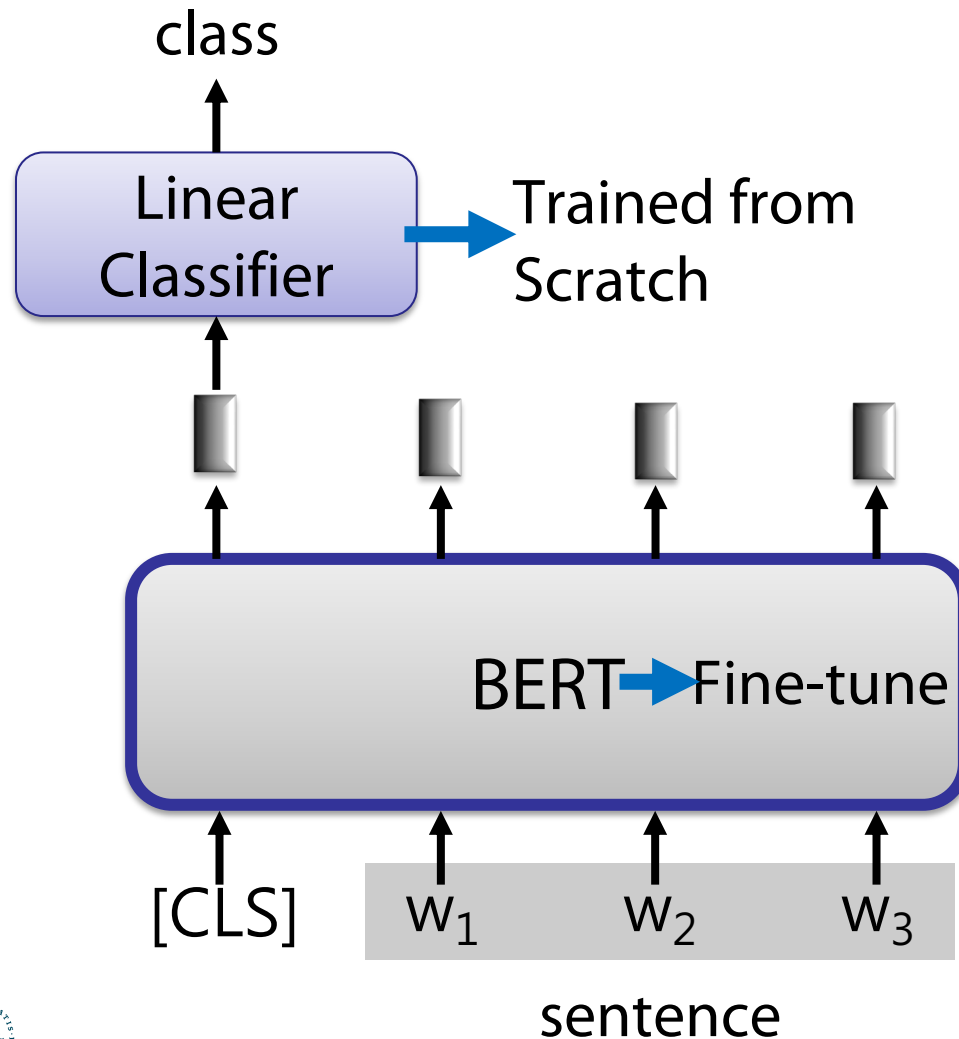


[CLS]: the position that outputs classification results

[SEP]: the boundary of two sentences

Approaches 1 and 2 are used at the same time.

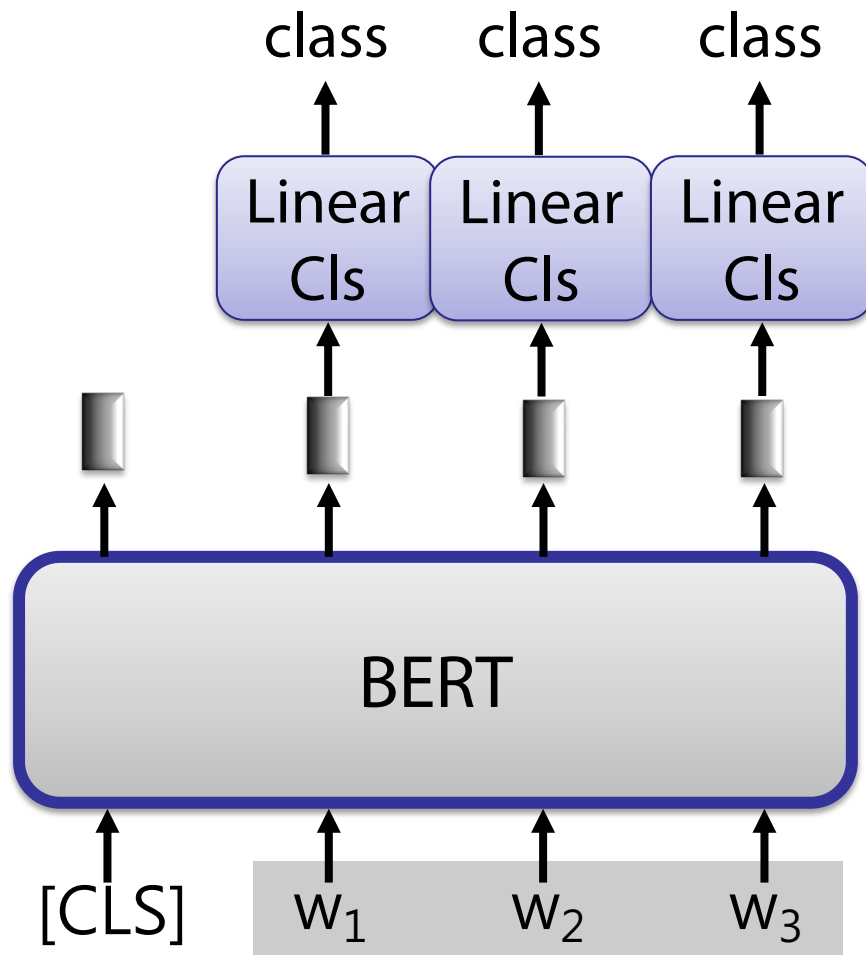
How to use BERT – Case 1



Input: single sentence,
output: class

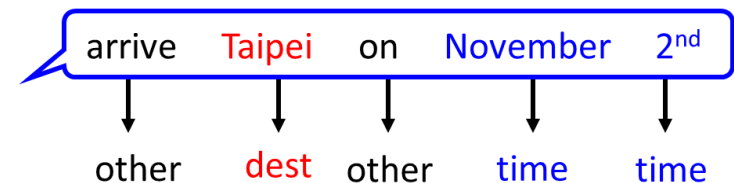
Example:
Sentiment analysis,
Document
Classification

How to use BERT – Case 2

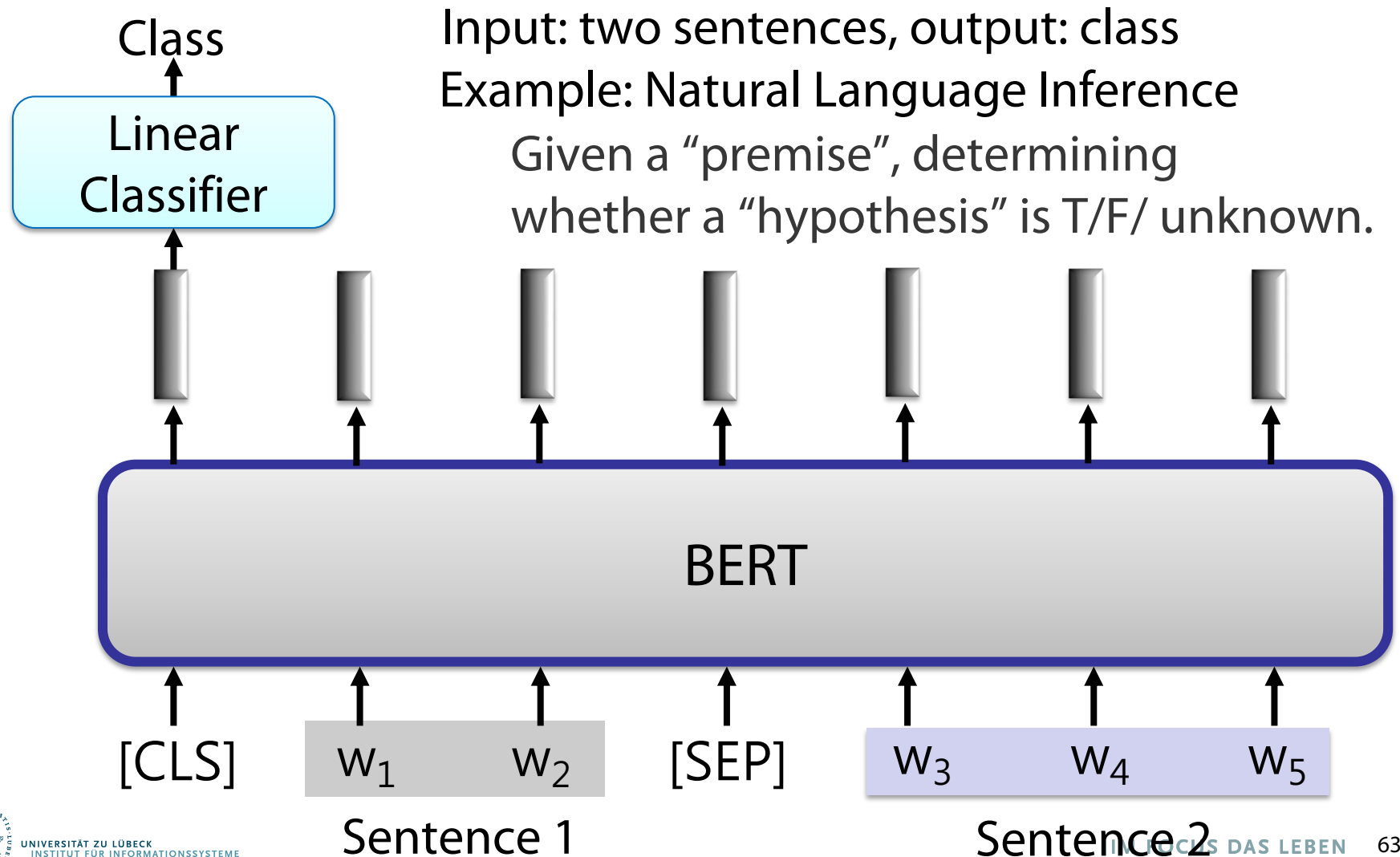


Input: single sentence,
output: class of each word

Example: Semantic role
labelling



How to use BERT – Case 3

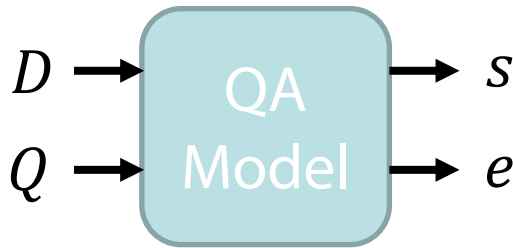


How to use BERT – Case 4

- Extraction-based Question Answering (QA) (E.g. SQuAD)

Document: $D = \{d_1, d_2, \dots, d_N\}$

Query: $Q = \{q_1, q_2, \dots, q_N\}$



output: two integers (s, e)

Answer: $A = \{q_s, \dots, q_e\}$

S=start, e=end

In meteorology, precipitation is any product of the condensation of 17 spheric water vapor that falls under **gravity**. The main forms of precipitation include drizzle, rain, sleet, snow, **grau-pel** 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 77 at 79 cations are called "showers".

What causes precipitation to fall?

gravity

$s = 17, e = 17$

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

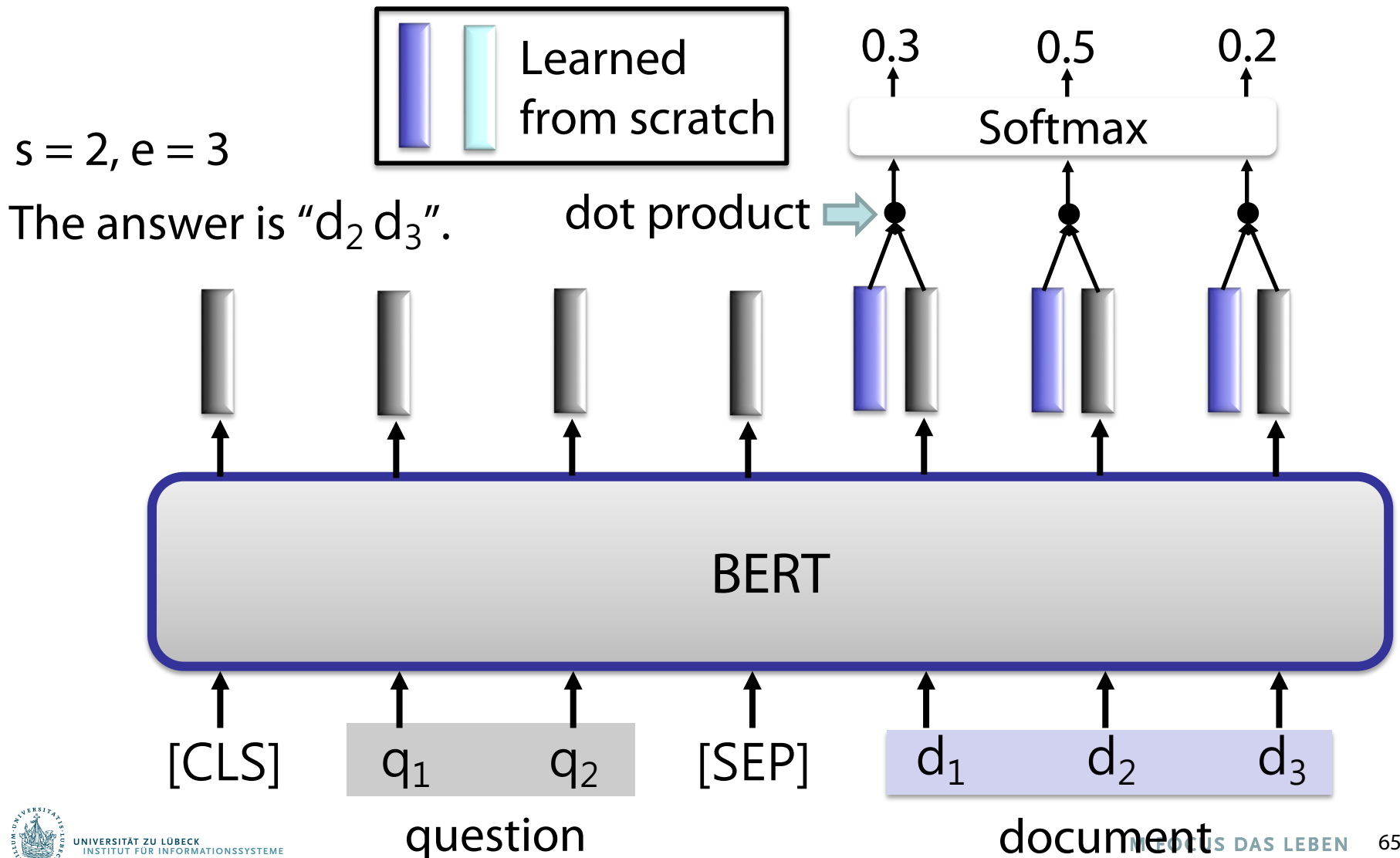
grau-pel

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

within a cloud

$s = 77, e = 79$

How to use BERT – Case 4

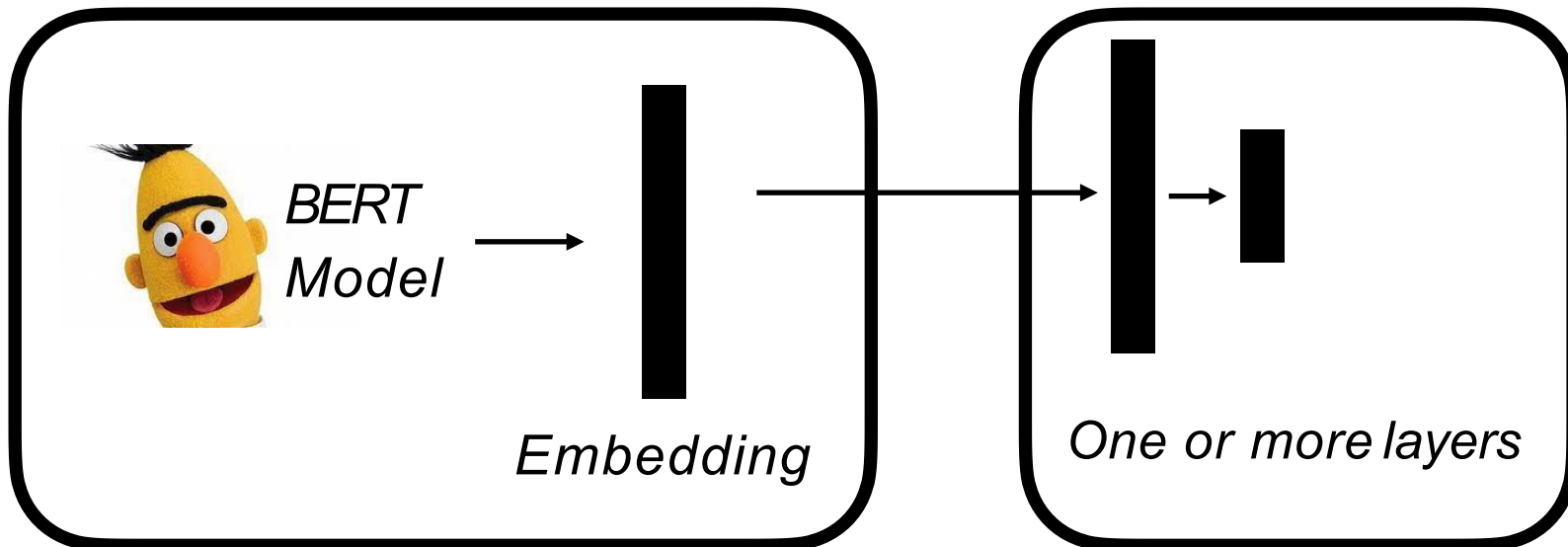


BERT Pre-Training & Fine Tuning

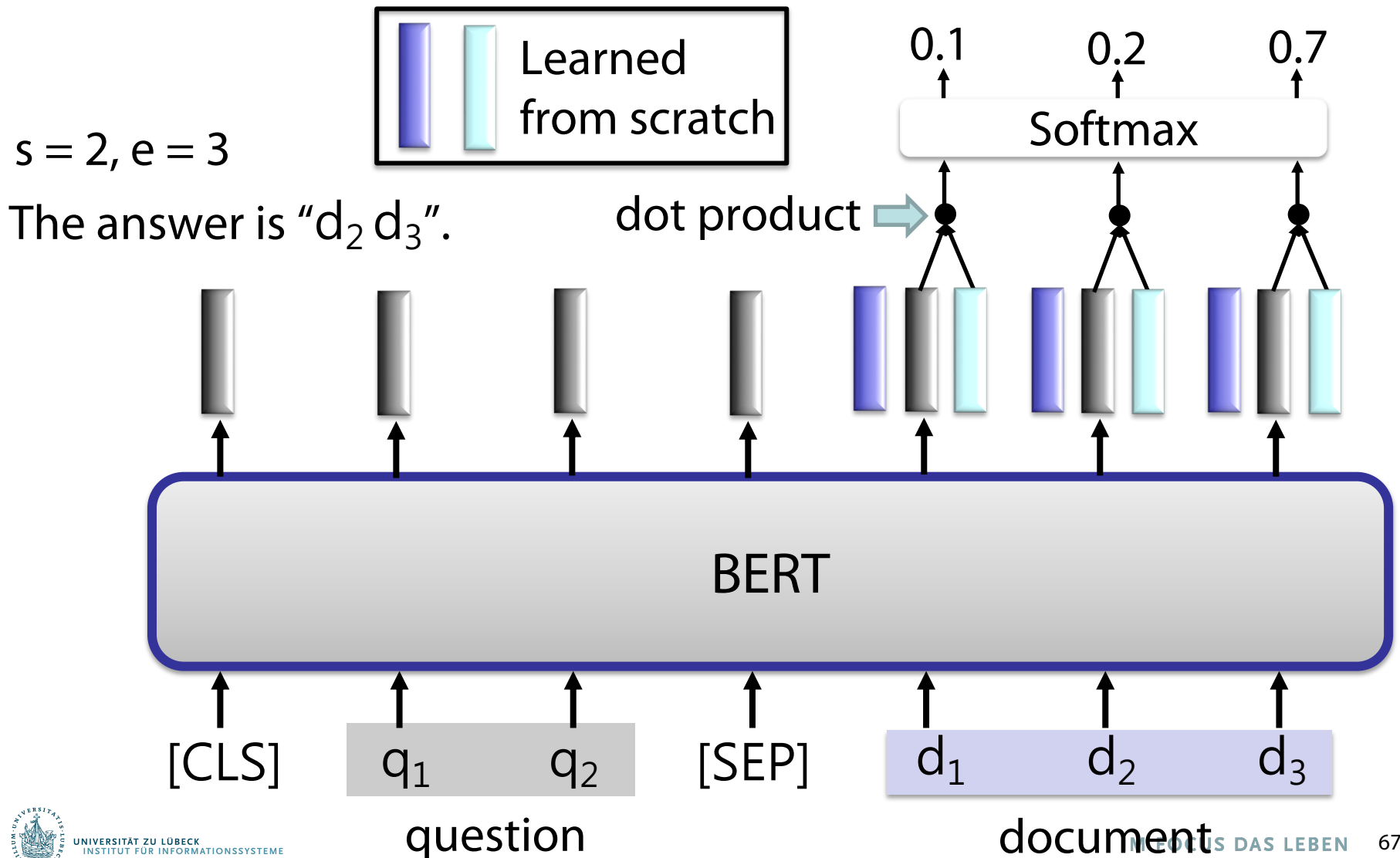
- Keep BERT frozen after pre-training
- Create BERT embeddings for labeled dataset for "downstream task" and train new model on these embeddings

1) Download BERT
pre-trained on large corpus
(in self-supervised fashion)

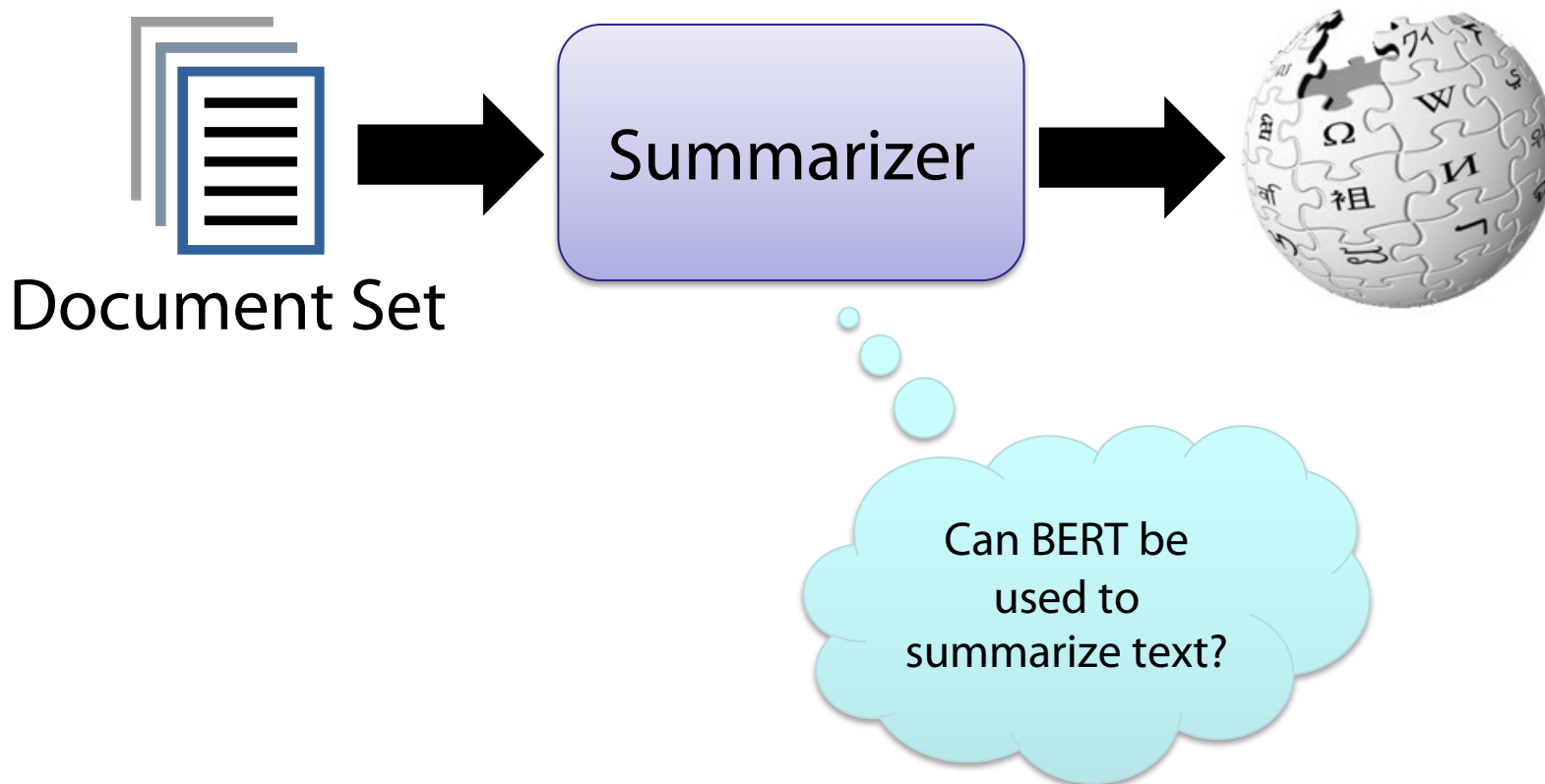
2) Feature-based training ("fine-tuning")
on target task
(supervised learning)



How to use BERT – Case 4



Example Application: Summarization



<https://arxiv.org/abs/1801.10198>

BERT as a Markov Random Field Language Model

- Wang et al. show that **BERT** (as described by Devlin et al., 2018) is essentially a **Markov random field language model**

Alex Wang, Kyunghyun Cho. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. Volume: In Proc. of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation, June **2019**.
<https://arxiv.org/abs/1902.04094>

Recap: From word2vec/ELMo via Transformers to BERT

- Language modeling is the “ultimate” NLP task
 - I.e., a perfect language model is also a perfect question answering/entailment/sentiment analysis model
 - Training a massive language model learns millions of latent features which are useful for these other NLP tasks
- E.g., for natural language inference
 - No internal “logical” representation
 - Use language directly to infer new propositions
 - ~~What kind of a thing is the meaning of a sentence?~~
 - What concrete phenomena do you have to deal with to understand a sentence?
- BERT was just a start – many extensions in the literature

Intelligent Agents

1d-CNNs LSTMs ELMo Transformers BERT GPT

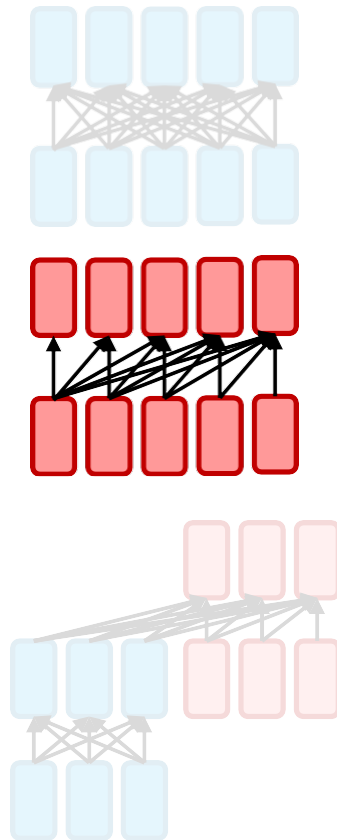
Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme



Model Pre-Training



- Encoder

- Bidirectional context

- Examples: BERT and its variants

- Decoder

- Language modeling; better for generation

- Example: GPT-2, GPT-3, [LaMDA](#)

- Encoder-Decoder

- Sequence-to-sequence model

- Examples: Transformer, BART, T5

LaMDA: Language Models for Dialog Applications. LaMDA is a family of Transformer- based neural language models specialized for dialog, which have up to 137B parameters and are pre-trained on 1.56T words of public dialog data and web text

GPT (Generative Pre-trained Transformer)

- Developed by OpenAI
- Unidirectional: trained to predict next word in a sentence

GPT (110 million parameters)

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

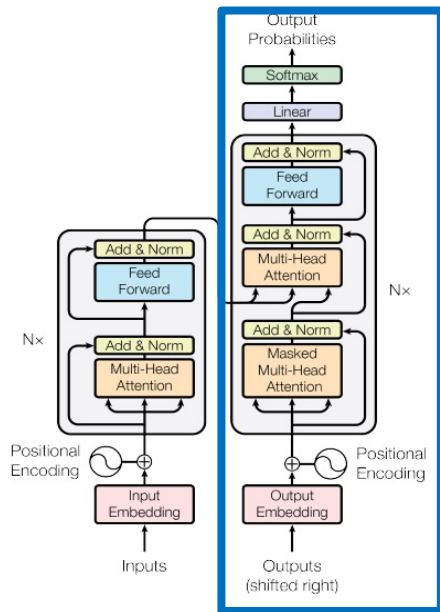
GPT-2 1.5 billion parameters)

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.
https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

GPT-3 (175 billion parameters)

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165. <https://arxiv.org/abs/2005.14165>

Generative Pre-Training (GPT)



Transformer Decoder

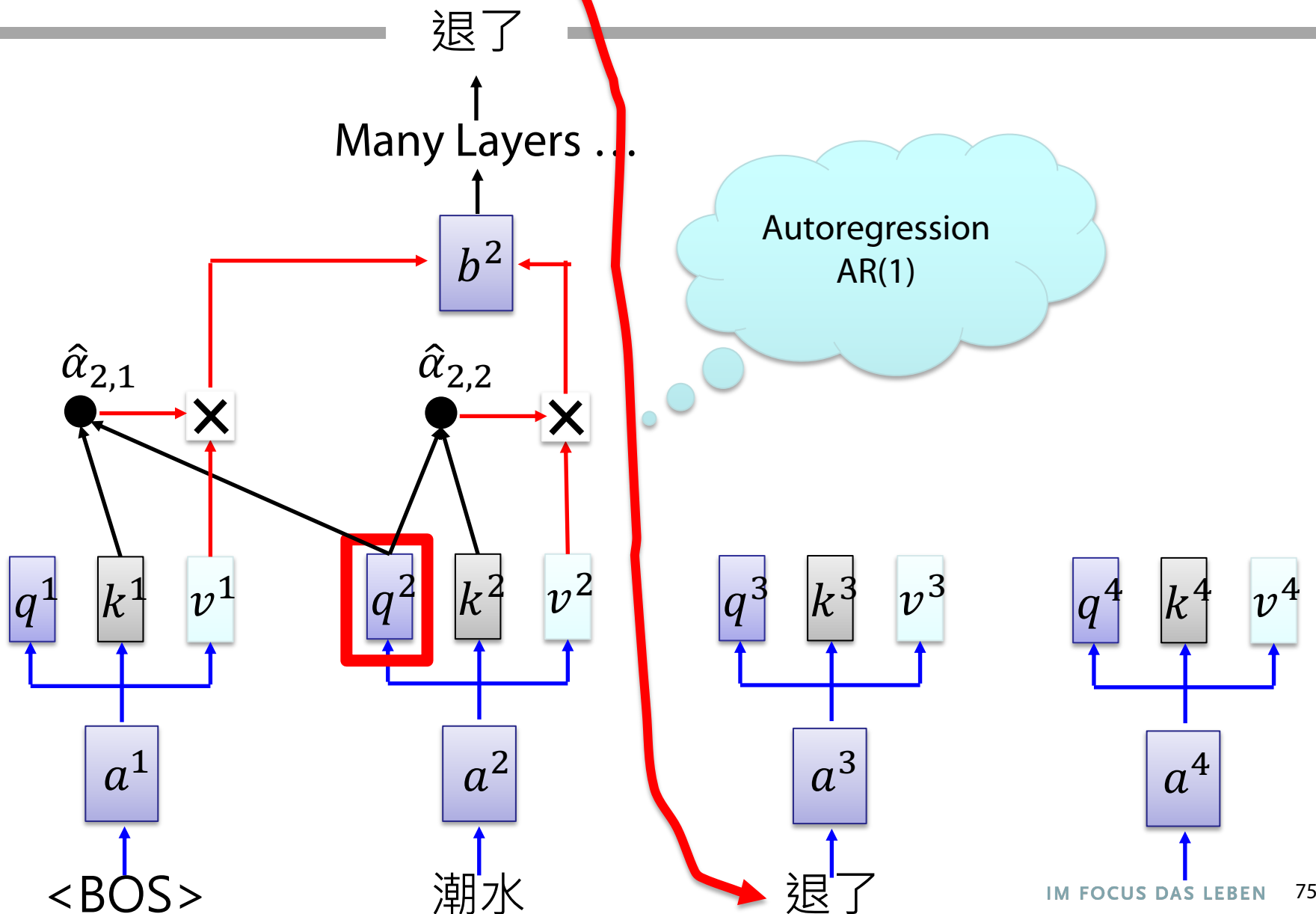


BERT
(340M)
ELMO
(94M)

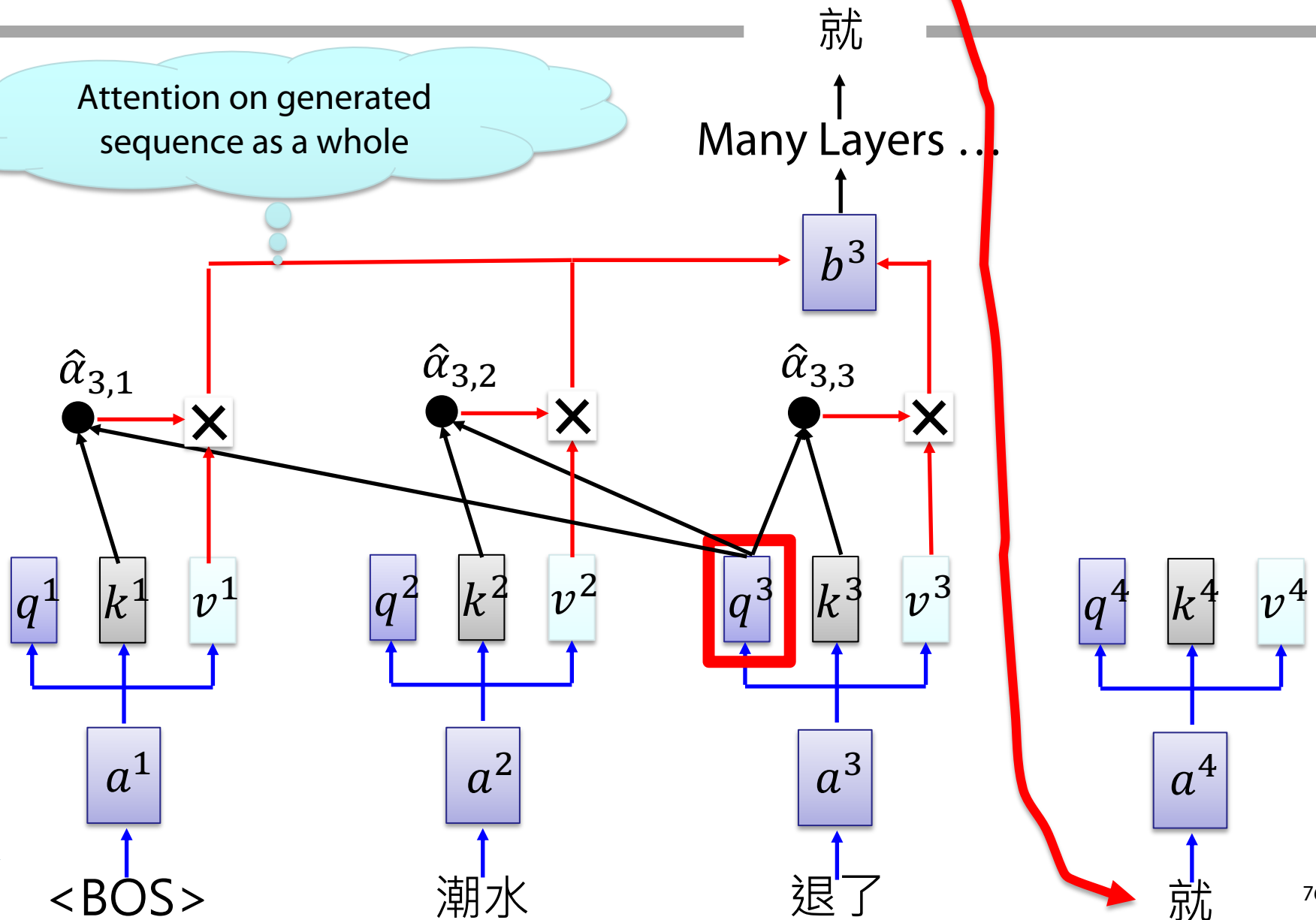


GPT-2
(1.5B)

Generative Pre-Training (GPT)



Generative Pre-Training (GPT)



Application: Summaries (Open AI)



ORIGINAL TEXT — 26,449 WORDS
SOURCE: PROJECT GUTENBERG

ALICE'S ADVENTURES IN WONDERLAND
Lewis Carroll

THE MILLENNIUM FULCRUM EDITION 3.0

CHAPTER I. Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversations?'

So she was considering in her own mind (as well as she could, for the hot day made her feel very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.

There was nothing so VERY remarkable in that; nor did Alice think it so VERY much out of the way to hear the Rabbit say to itself, 'Oh dear! Oh dear! I shall be late!' (when she thought it over afterwards, it occurred to her that she ought to have wondered at this, but at the time it all seemed quite natural); but when the Rabbit actually TOOK A WATCH OUT OF ITS WAISTCOAT-POCKET, and looked at it, and then hurried on, Alice started to her feet, for it flashed across her mind that she had never before seen a rabbit with either a waistcoat-pocket, or a watch to take out of it, and burning with curiosity, she ran across the field after it, and fortunately was just in time to see it pop down a large rabbit-hole under the hedge.

In another moment down went Alice after it, never once considering how in the world ~~she~~ was to get out again.

thousand miles down, I think-' (for several things of this sort in her life and though this was not a VERY good showing off her knowledge, as the White Rabbit, still it was good practice to say the right distance—but then I was not sure of my Longitude or Longitude I've got to?' (Alice had never heard of Longitude either, but thought it better to say.)

Presently she began again. 'I've never been THROUGH the earth! How funny to think among the people that walk with their heads Antipathies, I think-' (she was rather long listening, this time, as it didn't sound very good but I shall have to ask them what they think you know. Please, Ma'am, is this the way (and she tried to curtsy as she spoke) you're falling through the air! Do you think it?) 'And what an ignorant little girl!' No, it'll never do to ask: perhaps it's somewhere.'

Down, down, down. There was no end to it; she soon began talking again. 'Dinah was my cat, I should think!' (Dinah was the name of her saucer of milk at the time) 'I wish you were down here with me, but I'm afraid, but you might catch the mouse, you know. But do cats eat mice?' Alice began to get rather sleepy herself, in a dreamy sort of way, 'Do bats eat bats?' and sometimes 'Do bats eat

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15





The original text is divided into sections, and each section is summarized.

66 SUMMARIES — 6,024 WORDS

ALICE is bored sitting by her sister on the bank, and she's thinking about making a daisy chain when a white rabbit with pink eyes runs by. She's surprised to see a rabbit with a waistcoat pocket and a watch, and she follows it down a rabbit hole. She falls down a deep well, and as she's falling she takes note of the shelves and jars she passes. She thinks about how brave she'll seem when she gets home, and how she'll never tell anyone about her fall. She wonders how deep the well is and how many miles she's fallen.

ALICE thinks she's fallen a thousand miles, but she doesn't know what latitude or longitude she's at. She falls right through the earth and she'll emerge in a different world. She starts to get about walking hand in hand with her sister, but she's never eaten a bat. Suddenly she's dry leaves, and she's not hurt. She's in a moment.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

Alice's Adventures in Wonderland by Lewis Carroll



Section summaries are summarized again into higher-level summaries.

6 SUMMARIES — 830 WORDS

ALICE falls down a deep well and lands in a heap of sticks and leaves. She follows the White Rabbit down a long passage, but loses sight of it. She finds a bottle that says "Drink Me" on it and drinks it, shrinking to 10 inches tall. She grows to 9 feet tall and then shrinks again. She cries and grows larger again. A mouse appears and Alice asks it for help. The mouse agrees to tell Alice its story if she'll take it to the shore. Once there, the animals discuss how to get dry again. The Dodo suggests a Caucus-race to get dry. Everyone crowds around Alice asking for prizes. Alice pulls out a box of comfits and hands them out as prizes. The Mouse then asks Alice if she has a prize for herself, and she says she only has a thimble. The Dodo takes the thimble and presents it to Alice, saying they all beg her acceptance

ALICE grows larger after drinking the bottle. She stops growing before she reaches the table. She starts to miss being at home and then she follows the White Rabbit down the rabbit hole. She hears the White Rabbit say "Follow me" and she follows him. She realizes she's now much bigger than the White Rabbit. She is no longer afraid of it. Alice decides to follow the White Rabbit back to her normal size, and her size changes again. She goes away into the garden. She doesn't like the things, however. She sees a hedgehog and she asks him to hookah on a mushroom and asks him to grow her back to her normal size. The Caterpillar tells her that the mushroom will make her grow larger and the other mushroom will make her shorter. She starts eating the mushroom.

1 2 3 4 5 6



The summarizing process continues until a complete summary is achieved.

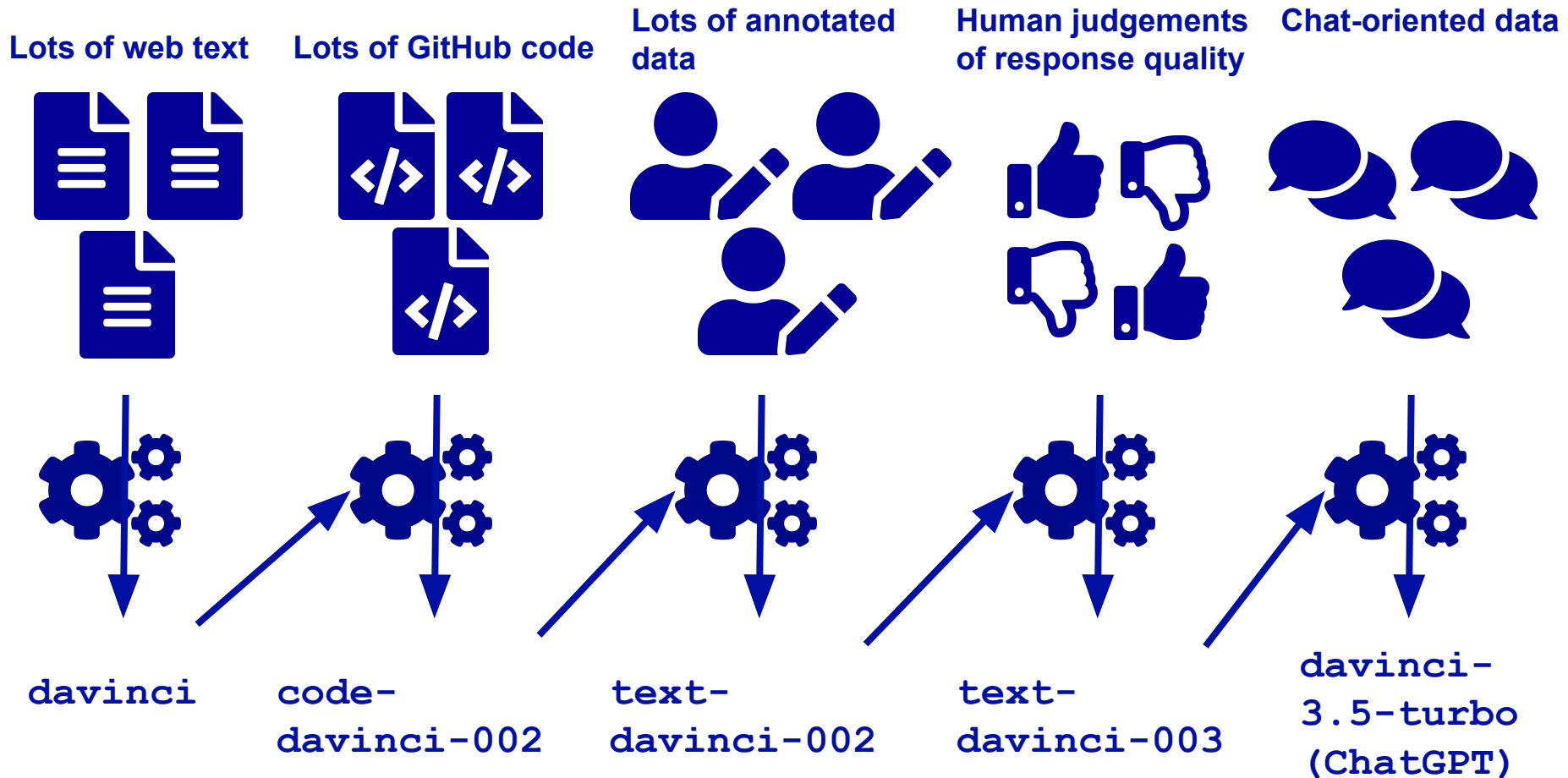
COMPLETE SUMMARY — 136 WORDS

ALICE falls down a rabbit hole and grows to giant size after drinking a mysterious bottle. She decides to focus on growing back to her normal size and finding her way into the garden. She meets the Caterpillar who tells her that one side of a mushroom will make her grow taller, the other side shorter. She eats the mushroom and returns to her normal size. Alice attends a party with the Mad Hatter and the March Hare. The Queen arrives and orders the execution of the gardeners for making a mistake with the roses. Alice saves them by putting them in a flowerpot. The King and Queen of Hearts preside over a trial. The Queen gets angry and orders Alice to be sentenced to death. Alice wakes up to find her sister by her side.

LLMs of the GPT family

- Generate term papers
 - Generate code
 - Generate Powerpoint presentations
 - Generate useful completions of texts in Word?
 - ...
-
- The latest version is GPT-4V

From Zero to ChatGPT

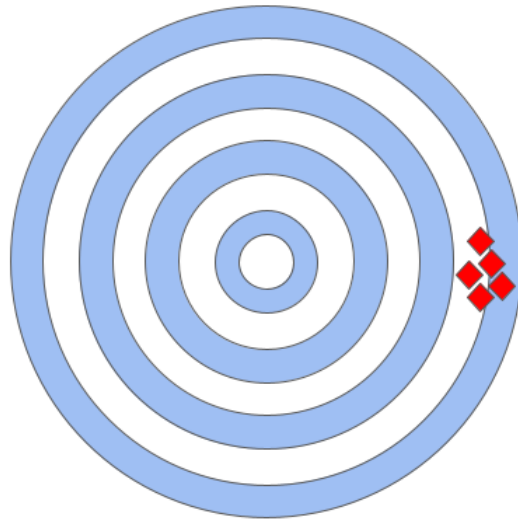


Compatibility vs. Alignment in LLMs

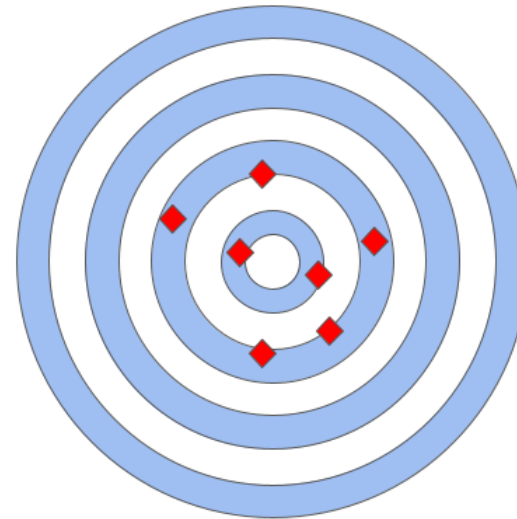
A model's *capability* is typically evaluated by **how well it is able to optimize its objective function**, the mathematical expression that defines the goal of the model

Alignment, on the other hand, is concerned with **what we actually want the model to do** versus what it is being trained to do

Models like the original GPT-3 are misaligned



High capability
Low alignment



Low capability
High alignment



Improving Language Model Behavior by Training on a Curated Dataset

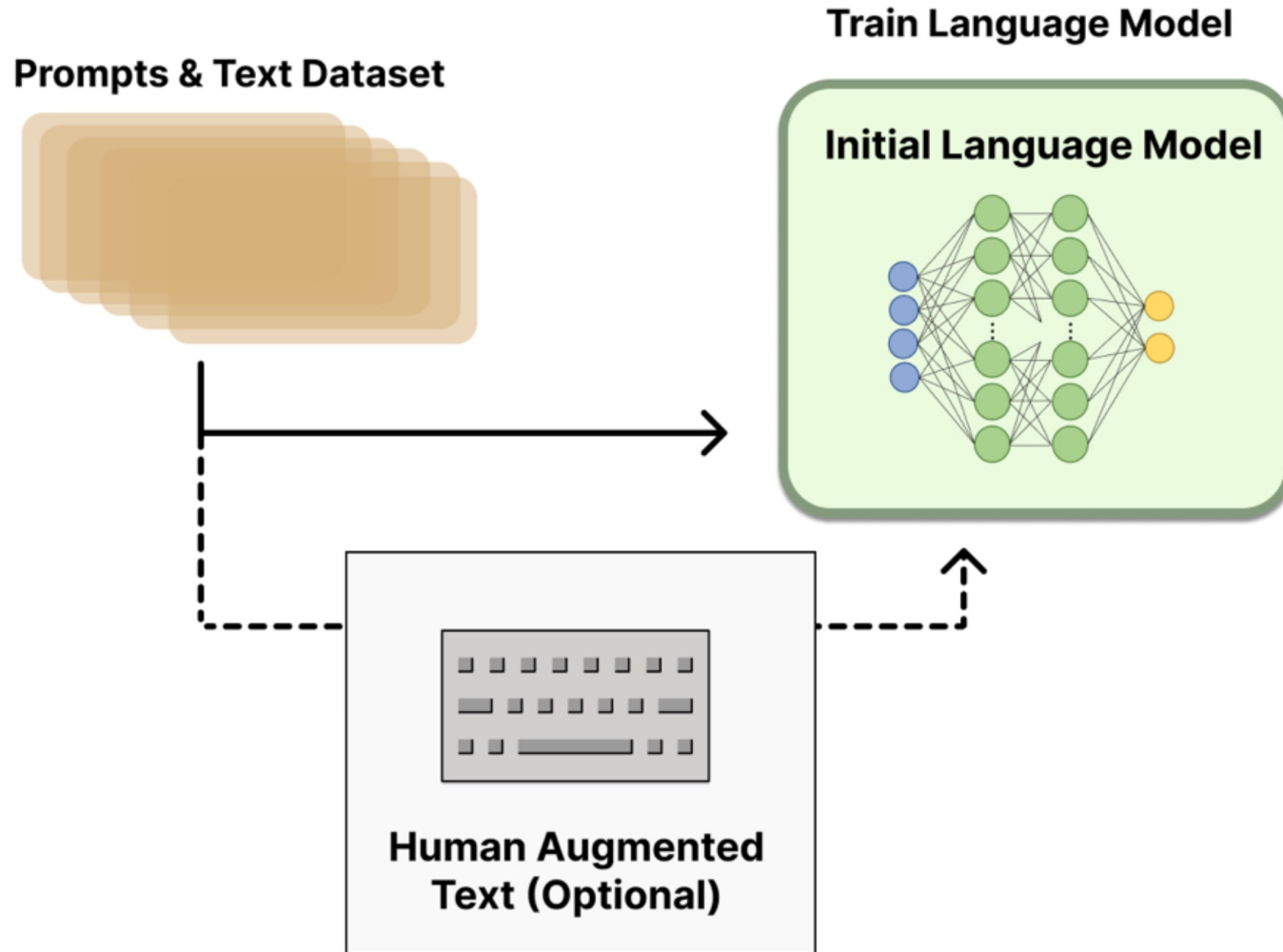
June 10, 2021
5 minute read

Our latest ~~research finds we can improve language model behavior with~~ respect to specific behavioral values by fine-tuning on a small, curated dataset.

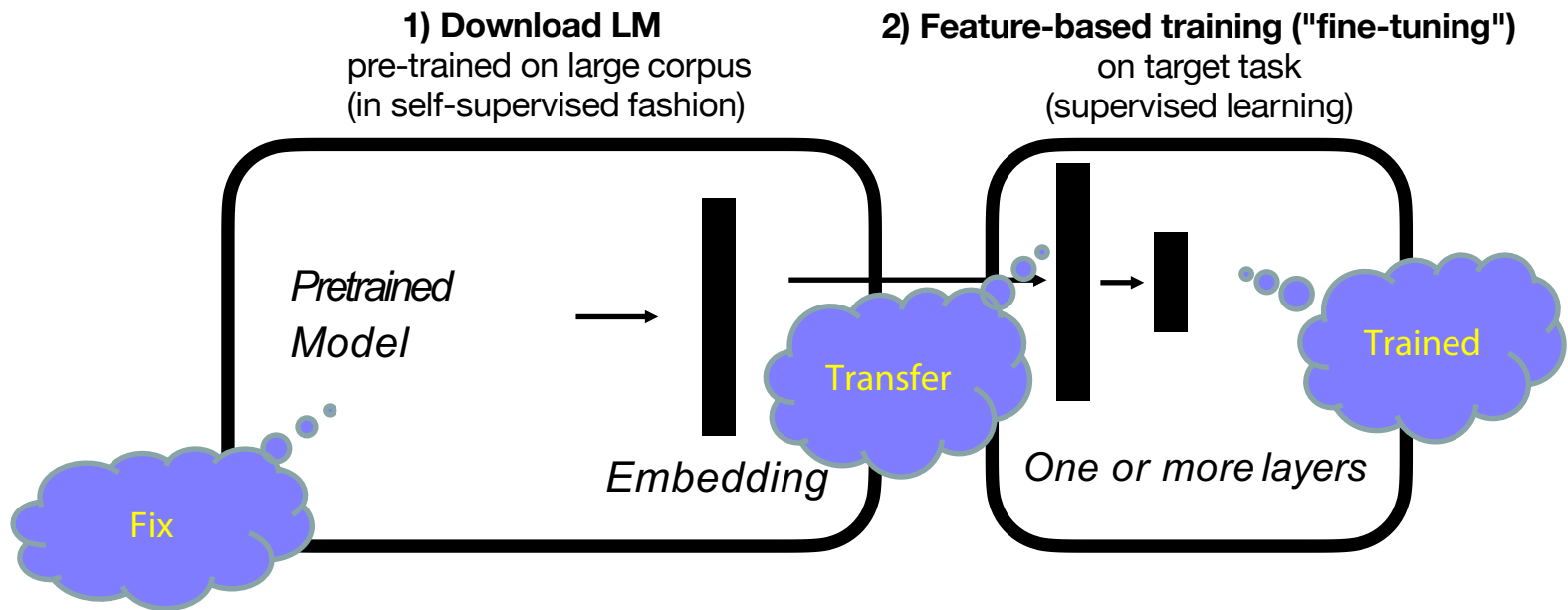
[READ PAPER](#)

We've found we can improve language model behavior with respect to specific behavioral values by fine-tuning on a curated dataset of <100 examples of those values. We also found that this process becomes more effective as models get larger. While the technique is still nascent, we're looking for OpenAI API users who would like to try it out and are excited to find ways to use these and other techniques in production use cases.

Train on Curated Dataset

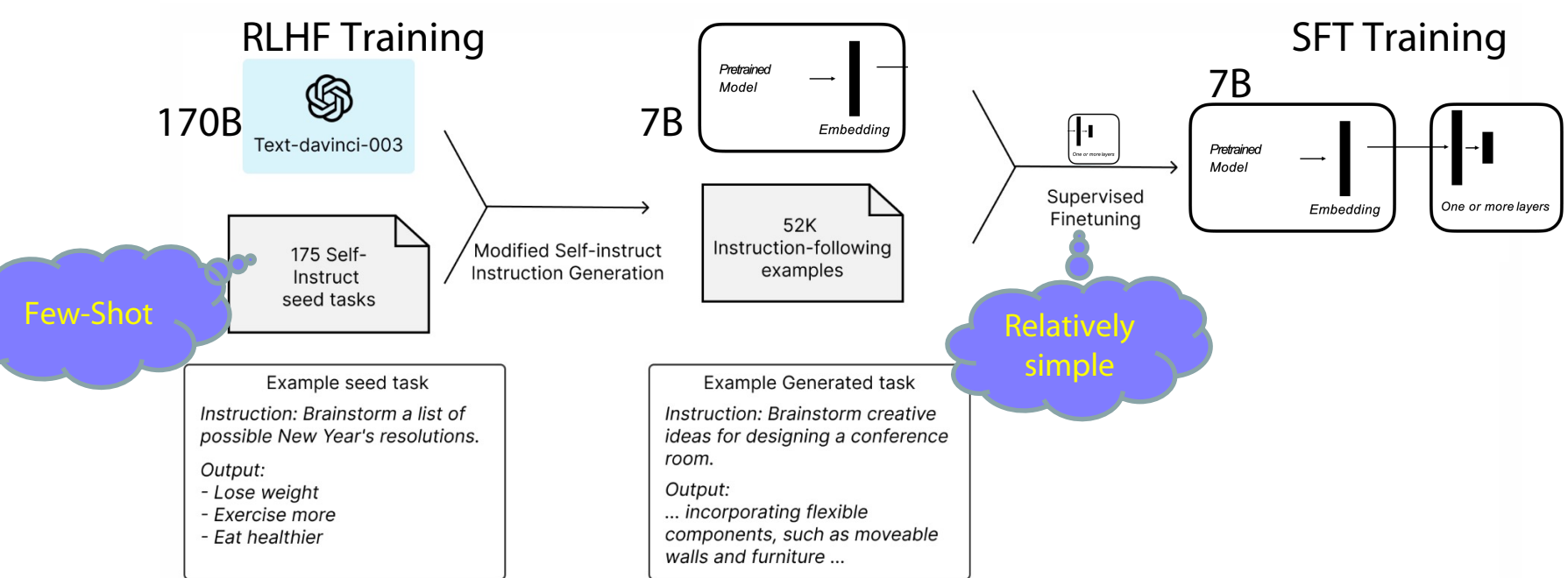


Pre-Training & Fine Tuning



Embedding & Training for Downstream Tasks

LLaMA vs Alpaca



Aligning Language Models to Follow Instructions

January 27, 2022
16 minute read

We've trained language models that are much better at following user intentions than GPT-3 while also making them more truthful and less toxic, using techniques developed through our alignment research. These *InstructGPT* models, which are trained with humans in the loop, are now deployed as the default language models on our API.

[READ PAPER](#)

[VIEW MODEL CARD](#)

InstructGPT is better than GPT-3 at following English instructions.

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

InstructGPT

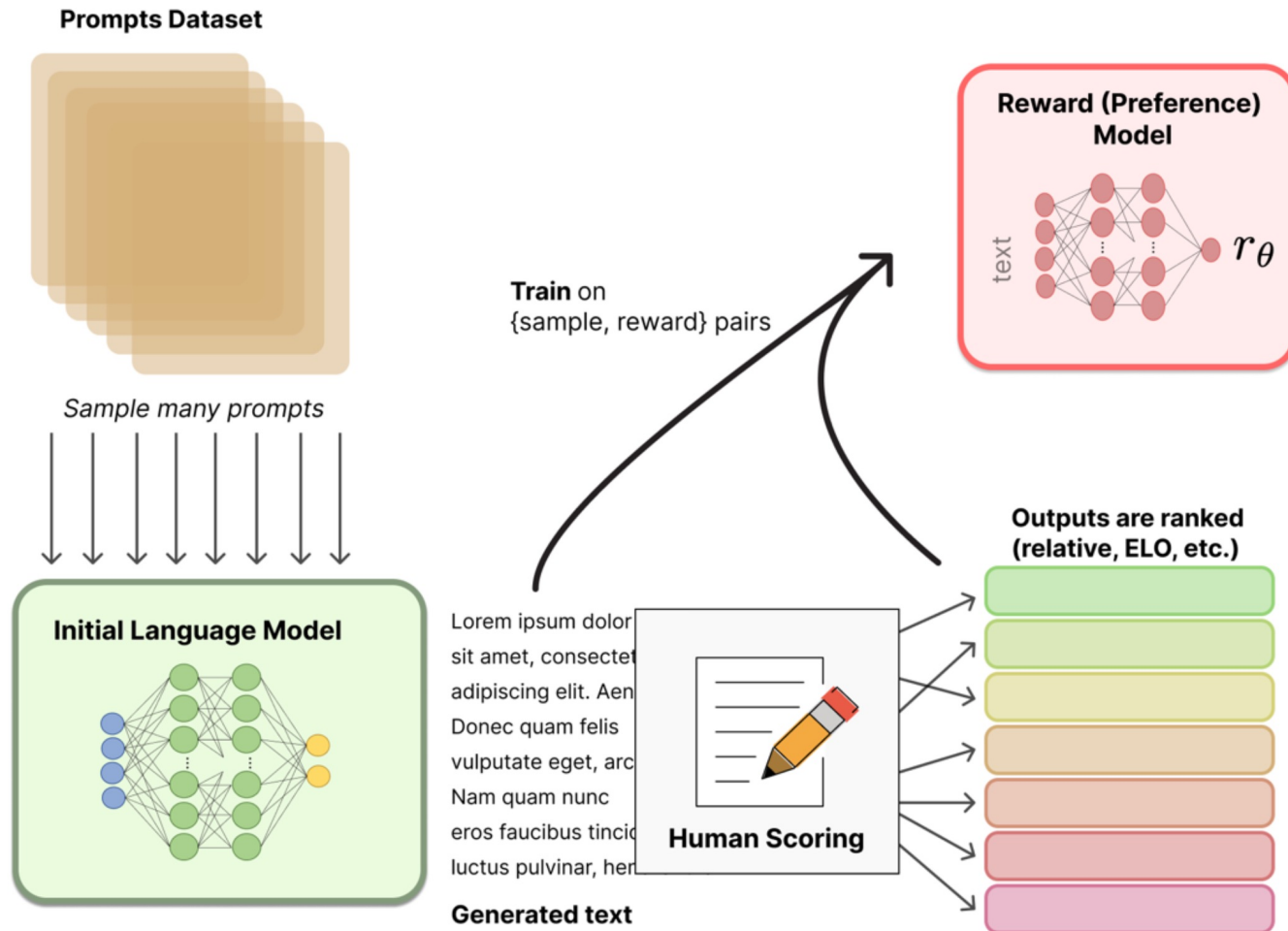
People went to the moon, and they took pictures of what they saw, and sent them back to the earth so we could all see them.



InstructGPT: Reinforcement Learning from Human Feedback (RLHF)

1. Pretraining a language model (LM),
 - OpenAI used a smaller version of GPT-3 for its first popular RLHF model, InstructGPT
2. Gathering data and training a reward model (RM, aka preference model), and
 - Get a model that takes in a sequence of text, and returns a scalar reward which should numerically represent the human preference
 - The training dataset of prompt-generation pairs for the RM is generated by sampling a set of prompts from a predefined dataset
3. Fine-tuning the LM with reinforcement learning

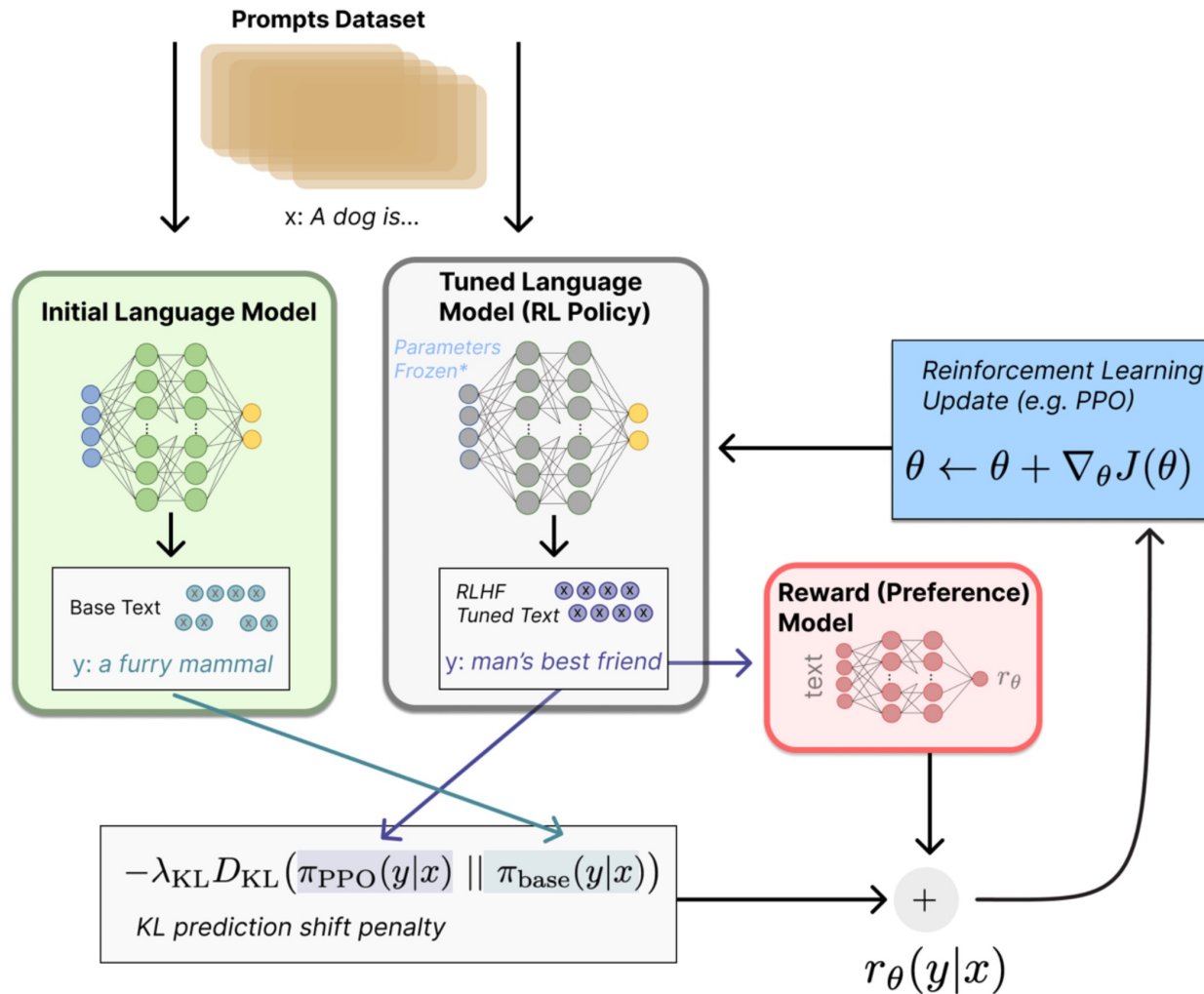
Train a Reward (Preference) Model



Reinforcement Learning

- **Policy** is a language model that takes in a prompt and returns a sequence of text (or just probability distributions over text).
- The **action space** of this policy is all the tokens corresponding to the vocabulary of the language model (often on the order of 50k tokens) and
- the **observation space** is the possible input token sequences, which is also quite large (size of vocabulary $^$ number of input tokens).
- The **reward function** is a combination of the preference model and a constraint on policy shift.
- Fine-tuning some or all of the parameters of a **copy of the initial LM** with a policy-gradient RL algorithm, Proximal Policy Optimization (PPO)
- Parameters of the LM are frozen because fine-tuning an entire 10B or 100B+ parameter model is prohibitively expensive (for more, see Low-Rank Adaptation (LoRA) for LMs or the Sparrow LM from DeepMind)

RLHF



Proximal Policy Optimization (PPO)

default reinforcement learning algorithm at OpenAI



Credits: Hung-yi Lee

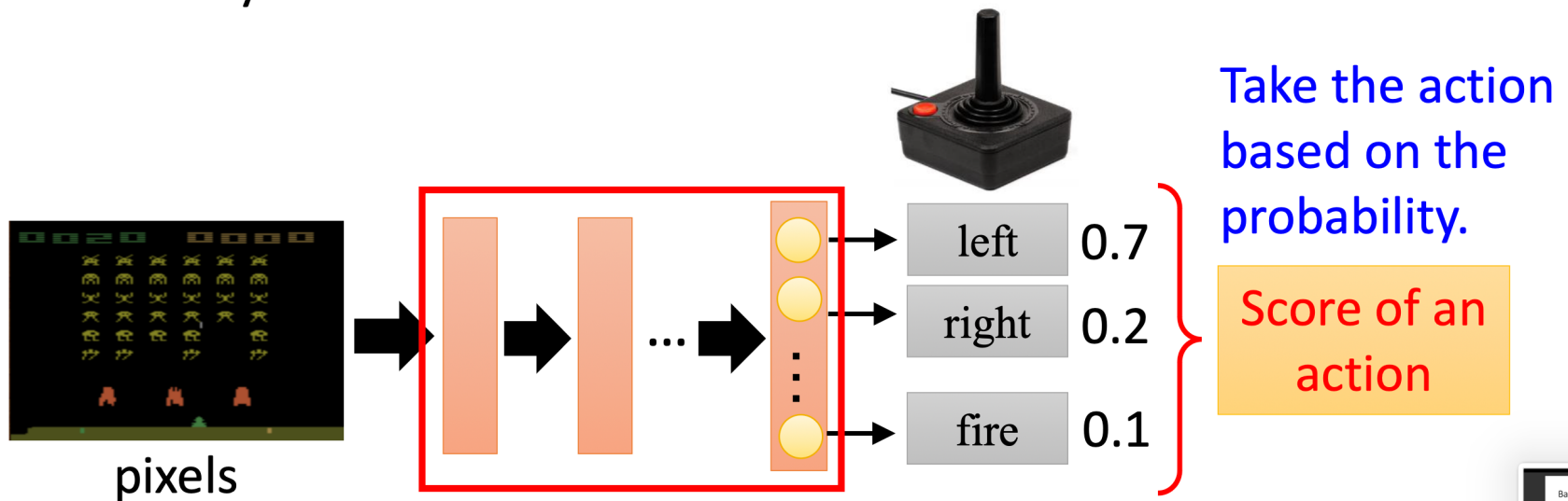
Basic Components



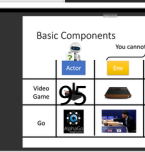
Credits: Hung-yi Lee

Policy of Actor

- Policy π is a network with parameter θ
 - Input: the observation of machine represented as a vector or a matrix
 - Output: each action corresponds to a neuron in output layer



Credits: Hung-yi Lee



Example: Playing Video Game

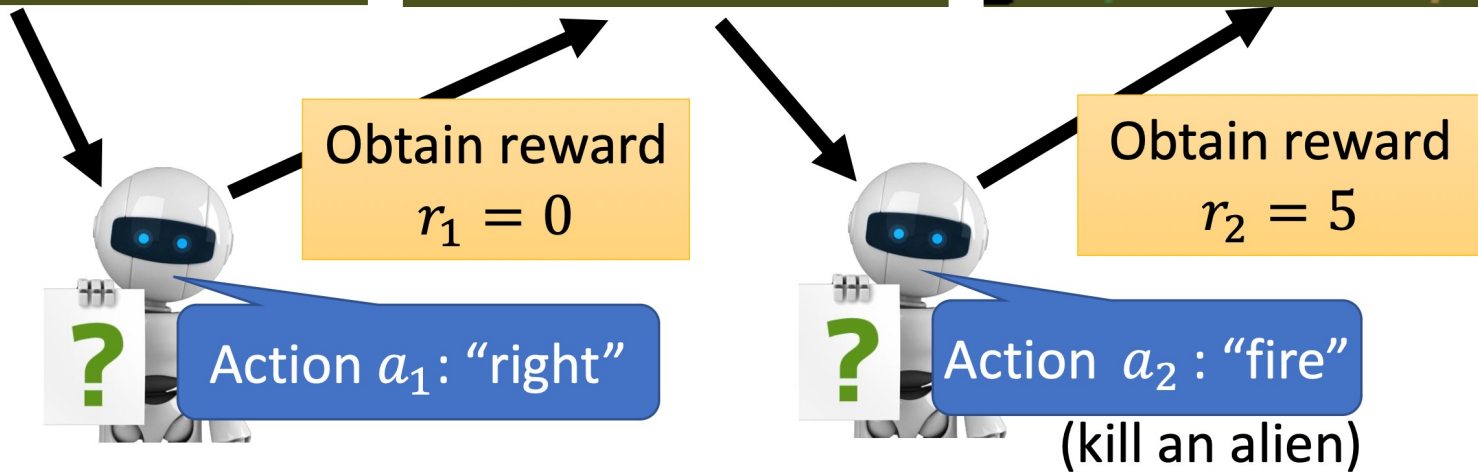
Start with
observation s_1



Observation s_2



Observation s_3



Credits: Hung-yi Lee

Example: Playing Video Game

Start with
observation s_1



Observation s_2



Observation s_3



After many turns



Obtain reward r_T

Action a_T

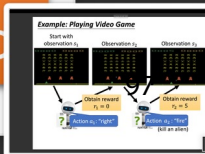
Credits: Hung-yi Lee

This is an episode.

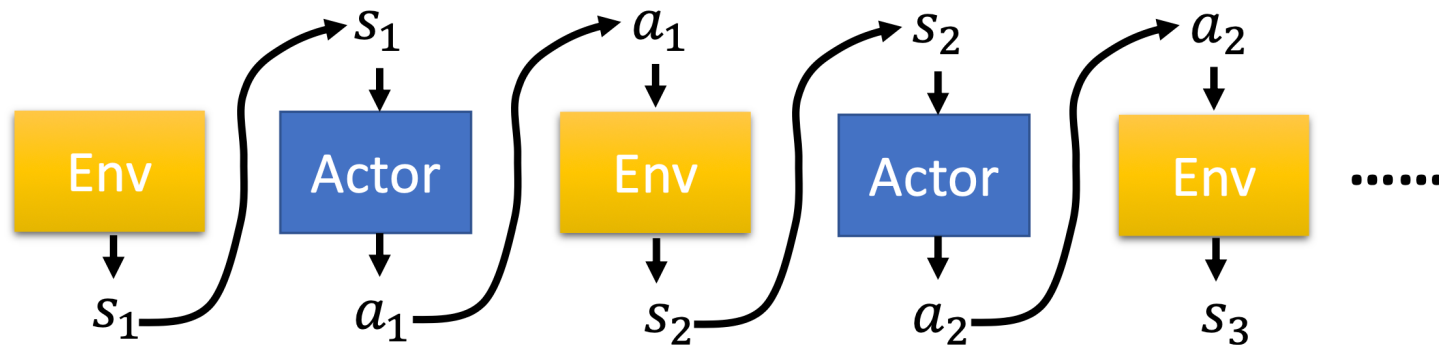
Total reward:

$$R = \sum_{t=1}^T r_t$$

We want the total
reward be maximized



Actor, Environment, Reward



Trajectory $\tau = \{s_1, a_1, s_2, a_2, \dots, s_T, a_T\}$

$p_{\theta}(\tau)$

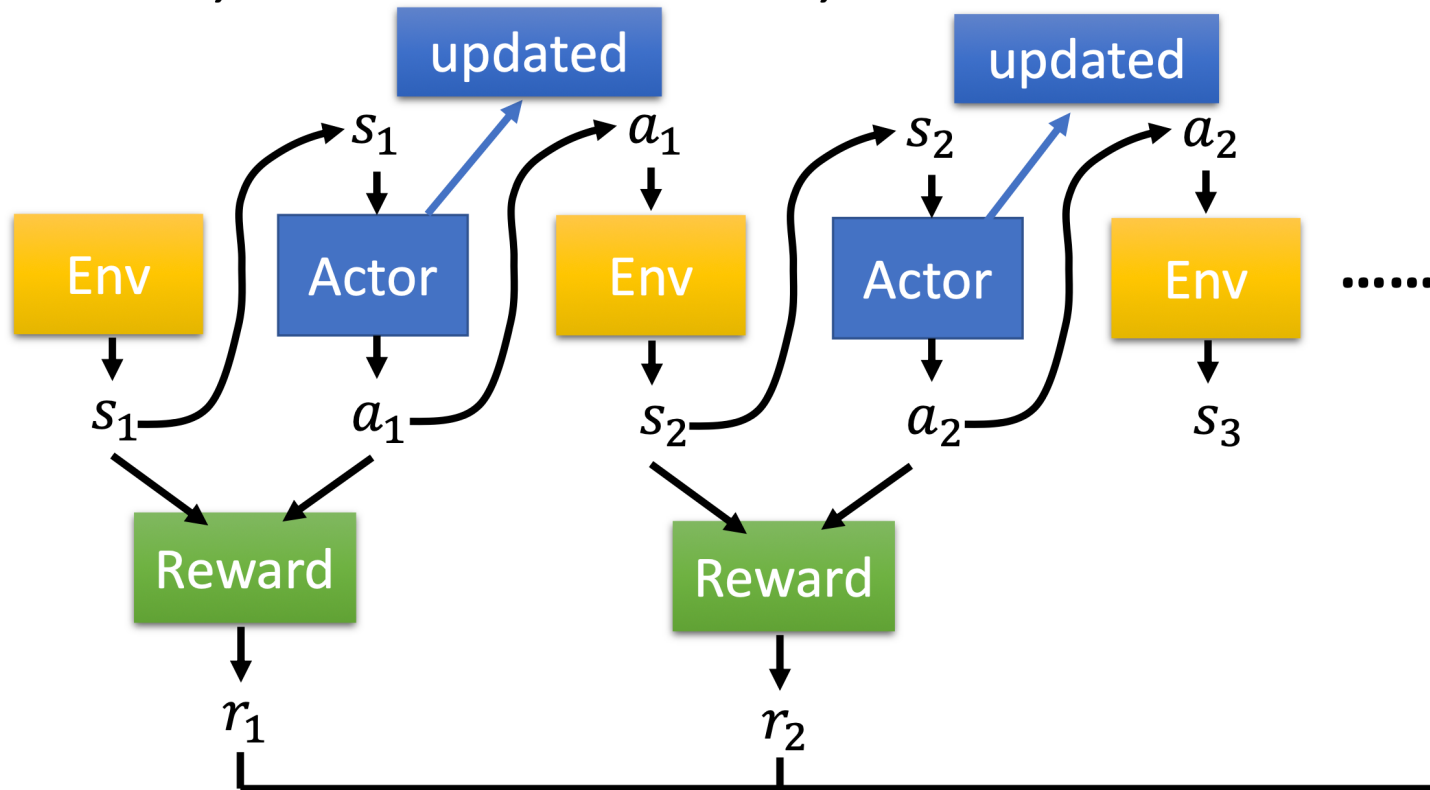
$$= p(s_1)p_{\theta}(a_1|s_1)p(s_2|s_1, a_1)p_{\theta}(a_2|s_2)p(s_3|s_2, a_2) \dots$$

$$= p(s_1) \prod_{t=1}^T p_{\theta}(a_t|s_t)p(s_{t+1}|s_t, a_t)$$

Credits: Hung-yi Lee



Actor, Environment, Reward



Expected Reward

$$\bar{R}_\theta = \sum_{\tau} R(\tau) p_\theta(\tau) = E_{\tau \sim p_\theta(\tau)} [R(\tau)]$$

Credits: Hung-yi Lee

$$R(\tau) = \sum_{t=1}^T r_t$$



Policy Gradient

$$\bar{R}_\theta = \sum_{\tau} R(\tau) p_\theta(\tau) \quad \nabla \bar{R}_\theta = ?$$

$$\nabla \bar{R}_\theta = \sum_{\tau} R(\tau) \nabla p_\theta(\tau) = \sum_{\tau} R(\tau) p_\theta(\tau) \frac{\nabla p_\theta(\tau)}{p_\theta(\tau)}$$

$R(\tau)$ do not have to be differentiable

It can even be a black box.

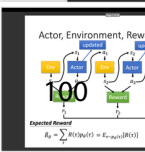
$$= \sum_{\tau} R(\tau) p_\theta(\tau) \nabla \log p_\theta(\tau)$$

$$\nabla f(x) = f(x) \nabla \log f(x)$$

$$= E_{\tau \sim p_\theta(\tau)} [R(\tau) \nabla \log p_\theta(\tau)] \approx \frac{1}{N} \sum_{n=1}^N R(\tau^n) \nabla \log p_\theta(\tau^n)$$

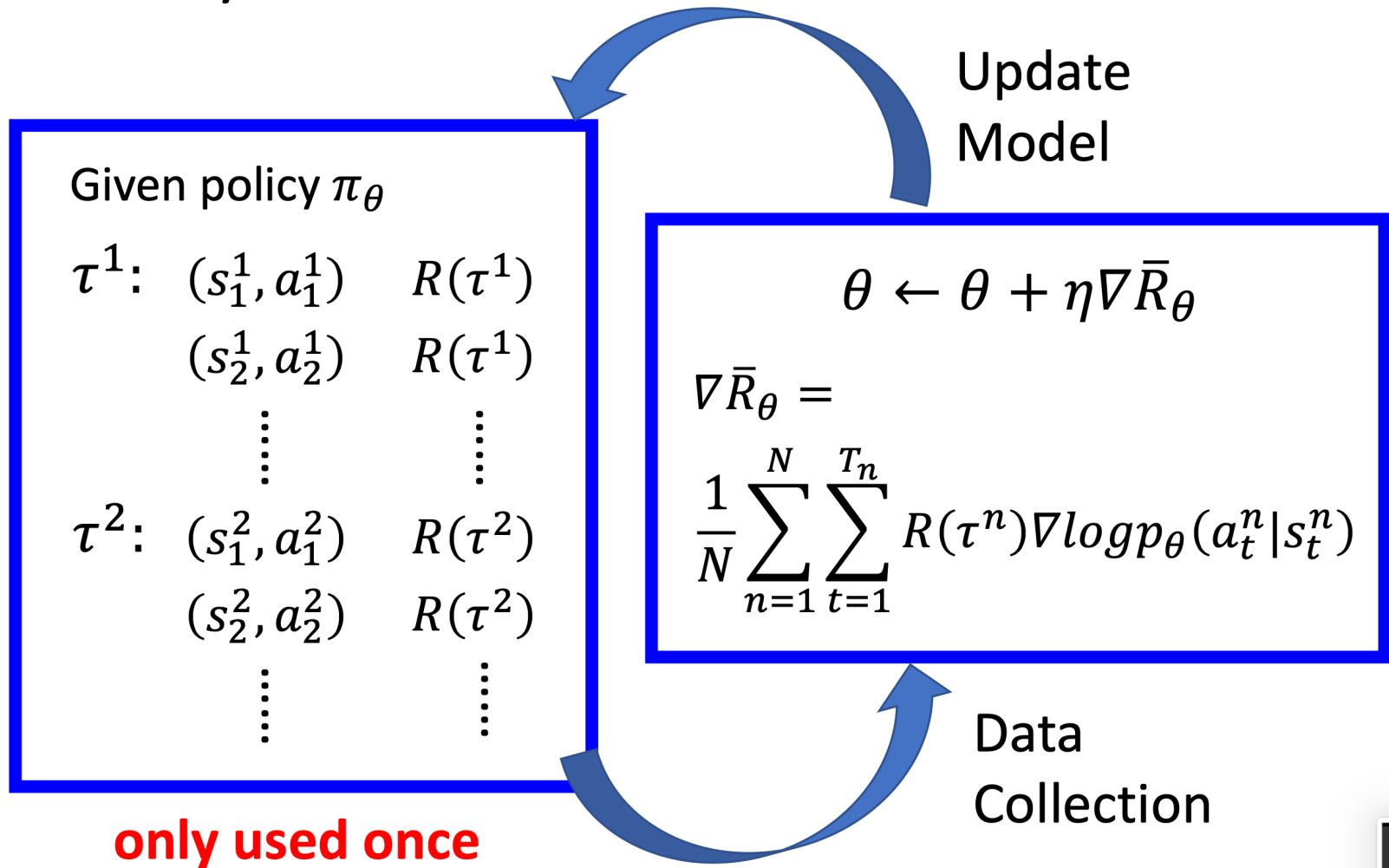
$$= \frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T_n} R(\tau^n) \nabla \log p_\theta(a_t^n | s_t^n)$$

Credits: Hung-yi Lee



Policy Gradient

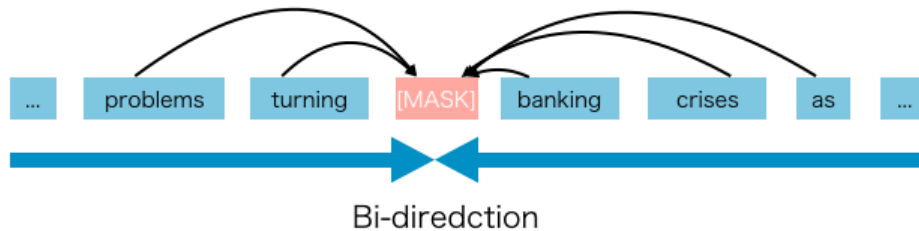
$$\nabla \bar{R}_\theta = E_{\tau \sim p_\theta(\tau)} [R(\tau) \nabla \log p_\theta(\tau)]$$



Credits: Hung-yi Lee

AE and AR

Autoencoding(AE) Language Modeling:



Can be implemented with self-attention

The AE language model aims to reconstruct the original data from **corrupted input**.

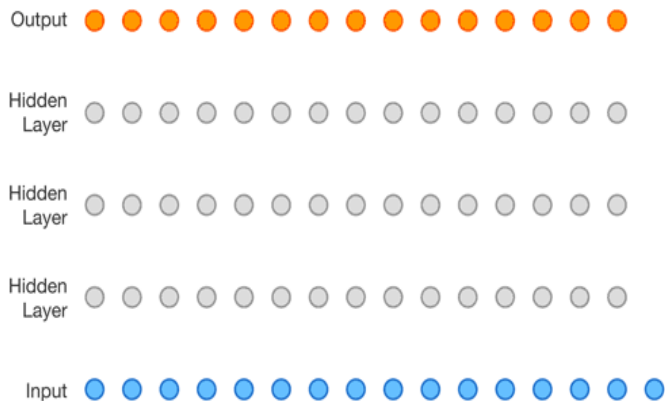
Corrupted input: The corrupted input means we use [MASK] to replace the original token

Example:

BERT

AE and AR

Autoregressive (AR)
language modeling:



Can also be implemented with attention
(GPT, not ELMO)

An autoregressive model's
output h_t at time t depends on not
just x_t , but also all x_s from previous
time steps.

given a text sequence $x = (x_1, \dots, x_T)$,
AR language modeling factorizes the
likelihood into a forward
product. $p(x) = \prod p(x_t | x_{<t})$

Examples:

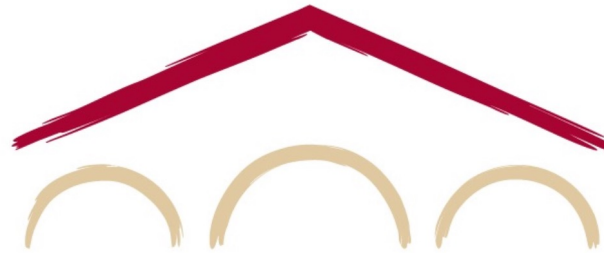
GPT, ELMO

GPT: Temperature&Top-p-sampling



Natural Language Processing with Deep Learning

CS224N/Ling284



Xiang Lisa Li

Lecture 12: Neural Language Generation

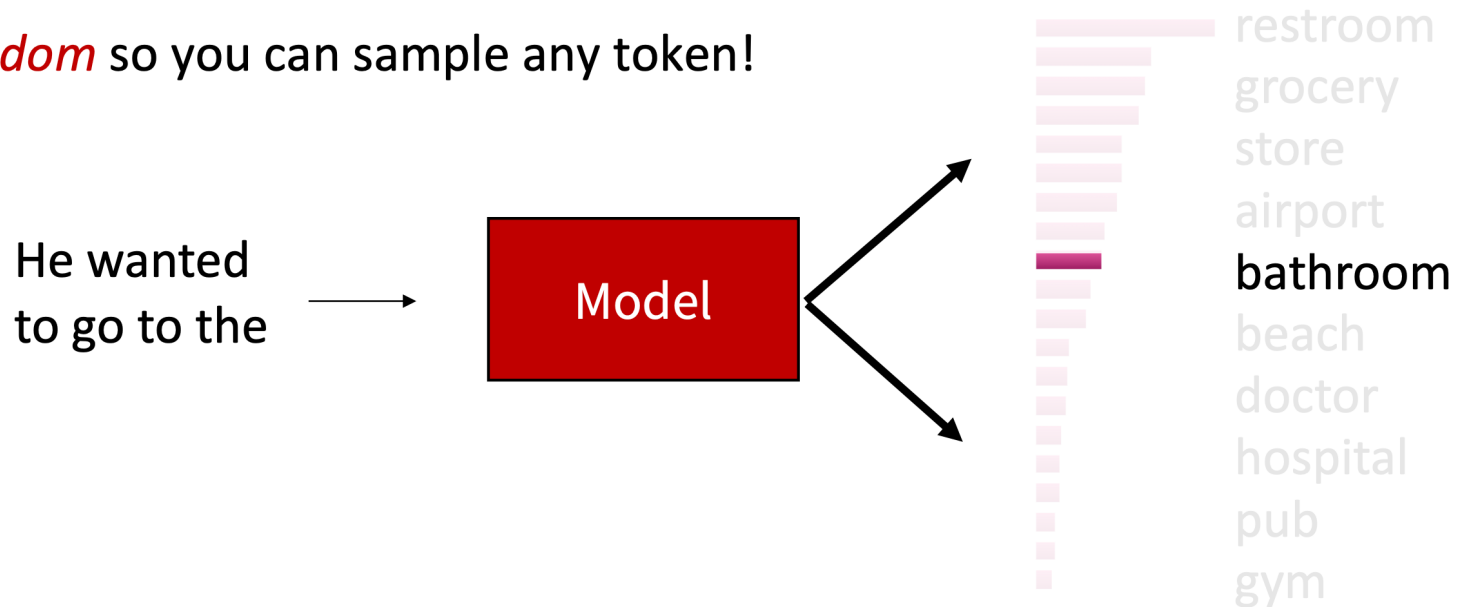
Adapted from slides by Antoine Bosselut and Chris Manning

Time to get random : Sampling!

- Sample a token from the distribution of tokens

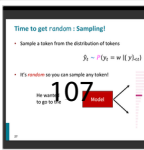
$$\hat{y}_t \sim P(y_t = w | \{y\}_{<t})$$

- It's *random* so you can sample any token!



Decoding: Top- k sampling

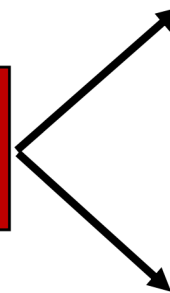
- Problem: Vanilla sampling makes every token in the vocabulary an option
 - Even if most of the **probability mass** in the distribution is over a limited set of options, the tail of the distribution could be very long and in aggregate have considerable mass (statistics speak: we have “**heavy tailed**” distributions)
 - Many tokens are probably *really wrong* in the current context
 - For these wrong tokens, we give them *individually* a tiny chance to be selected.
 - But because there are many of them, we still give them *as a group* a high chance to be selected.
- Solution: Top- k sampling
 - Only sample from the top k tokens in the probability distribution



Decoding: Top- k sampling

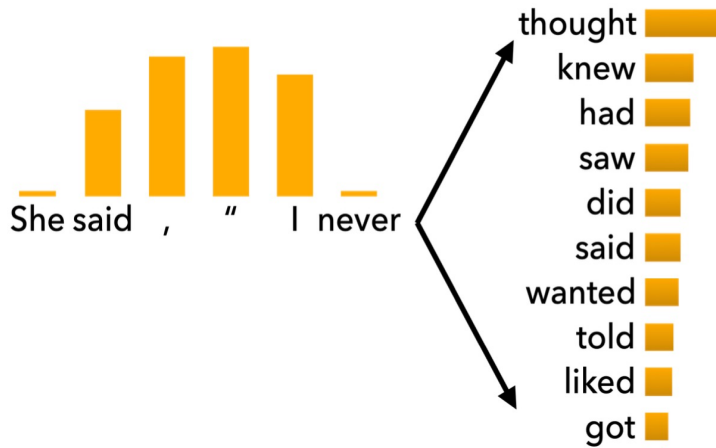
- Solution: Top- k sampling
 - Only sample from the top k tokens in the probability distribution
 - Common values are $k = 50$ (*but it's up to you!*)

He wanted
to go to the

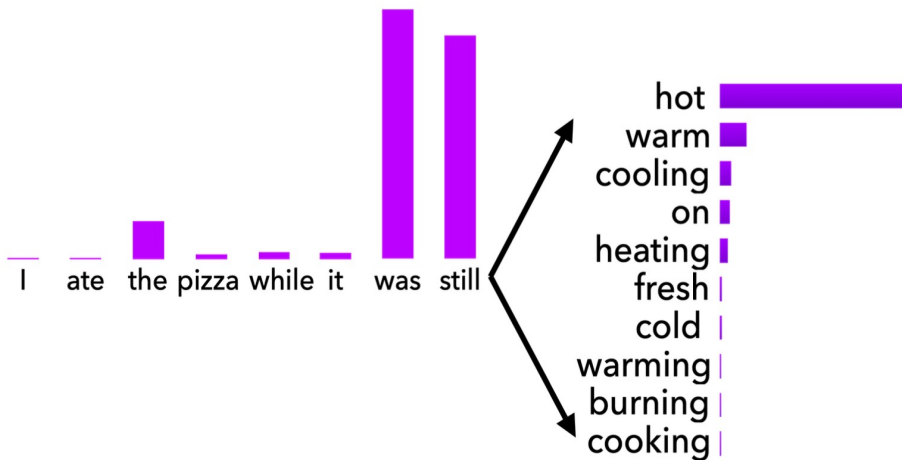


- Increase k yields more **diverse**, but **risky** outputs
- Decrease k yields more **safe** but **generic** outputs

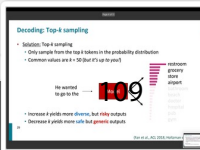
Issues with Top-k sampling



Top-k sampling can cut off too **quickly!**



Top-k sampling can also cut off too **slowly!**



Decoding: Top- p (nucleus) sampling

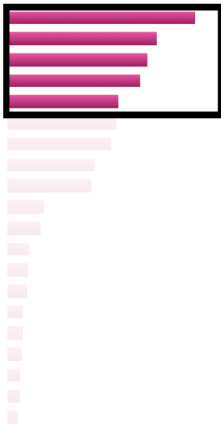
- Problem: The probability distributions we sample from are dynamic
 - When the distribution P_t is flatter, a limited k removes many viable options
 - When the distribution P_t is peakier, a high k allows for too many options to have a chance of being selected
- Solution: Top- p sampling
 - Sample from all tokens in the top p cumulative probability mass (i.e., where mass is concentrated)
 - Varies k depending on the uniformity of P_t



Decoding: Top- p (nucleus) sampling

- Solution: Top- p sampling
 - Sample from all tokens in the top p cumulative probability mass (i.e., where mass is concentrated)
 - Varies k depending on the uniformity of P_t

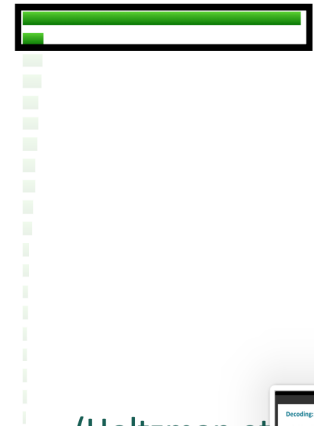
$$P_t^1(y_t = w | \{y\}_{<t})$$



$$P_t^2(y_t = w | \{y\}_{<t})$$



$$P_t^3(y_t = w | \{y\}_{<t})$$

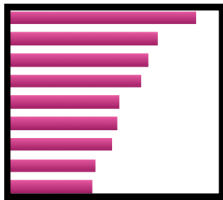


(Holtzman et al.)

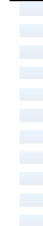
Decoding: More to go

- Typical Sampling (Meister et al. 2022)
 - Reweights the score based on the entropy of the distribution.
- Epsilon Sampling (Hewitt et al. 2022)
 - Set a threshold for lower bounding valid probabilities.

$$P_t^1(y_t = w | \{y\}_{<t})$$



$$P_t^2(y_t = w | \{y\}_{<t})$$



$$P_t^3(y_t = w | \{y\}_{<t})$$



Scaling randomness: Temperature

- Recall: On timestep t , the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^{|V|}$

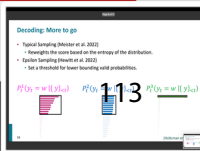
$$P_t(y_t = w) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- You can apply a *temperature hyperparameter* τ to the softmax to rebalance P_t :

$$P_t(y_t = w) = \frac{\exp(S_w/\tau)}{\sum_{w' \in V} \exp(S_{w'}/\tau)}$$

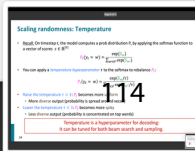
- Raise the temperature $\tau > 1$: P_t becomes more uniform
 - **More** diverse output (probability is spread around vocab)
- Lower the temperature $\tau < 1$: P_t becomes more spiky
 - **Less** diverse output (probability is concentrated on top words)

Temperature is a hyperparameter for decoding:
It can be tuned for both beam search and sampling.



Improving Decoding: Re-ranking

- Problem: What if I decode a bad sequence from my model?
- Decode a bunch of sequences
 - 10 candidates is a common number, but it's up to you
- Define a score to approximate quality of sequences and **re-rank by this score**
 - Simplest is to use (low) **perplexity!**
 - Careful! Remember that **repetitive utterances** generally get low perplexity.
 - Re-rankers can score a **variety of properties**:
 - style (Holtzman et al., 2018), discourse (Gabriel et al., 2021), entailment/factuality (Goyal et al., 2020), logical consistency (Lu et al., 2020), and many more ...
 - Beware poorly-calibrated re-rankers
 - Can compose multiple re-rankers together.



Decoding: Takeaways

- Decoding is still a challenging problem in NLG – **there's a lot more work to be done!**
- Different decoding algorithms can allow us to inject biases that encourage different properties of coherent natural language generation
- Some of the most **impactful advances** in NLG of the last few years have come from **simple** but **effective** modifications to decoding algorithms

Summarization

- **Extractive Text Summarization**

- The **traditional** method with the main objective to identify the significant sentences of the text and add them to the summary. Note that the summary obtained contains **exact sentences from the original text data**.
- Can be done with encoder (e.g., **BERT**)

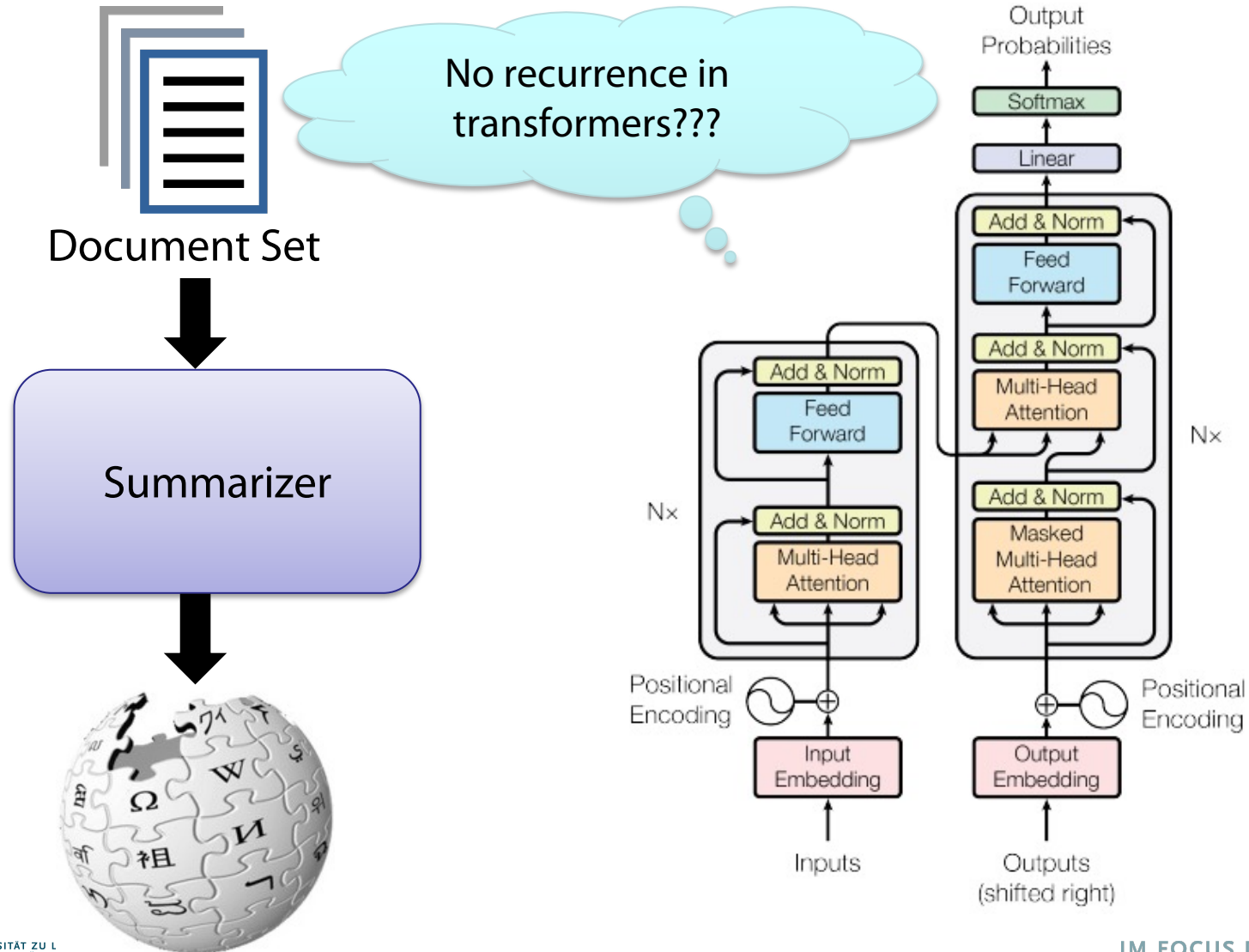
- **Abstractive Text Summarization**

- The **advanced** method, with the approach to identify the important sections, interpret the context and reproduce the text in a new way. This ensures that the core information is conveyed through the shortest text possible. Note that here, the sentences, in summary, **are generated by the model, not just extracted from the original text data**.
- Need Decoder (e.g., **GPT-x, PEGASUS**)

<https://medium.com/analytics-vidhya/text-summarization-using-bert-gpt2-xlnet-5ee80608e961>

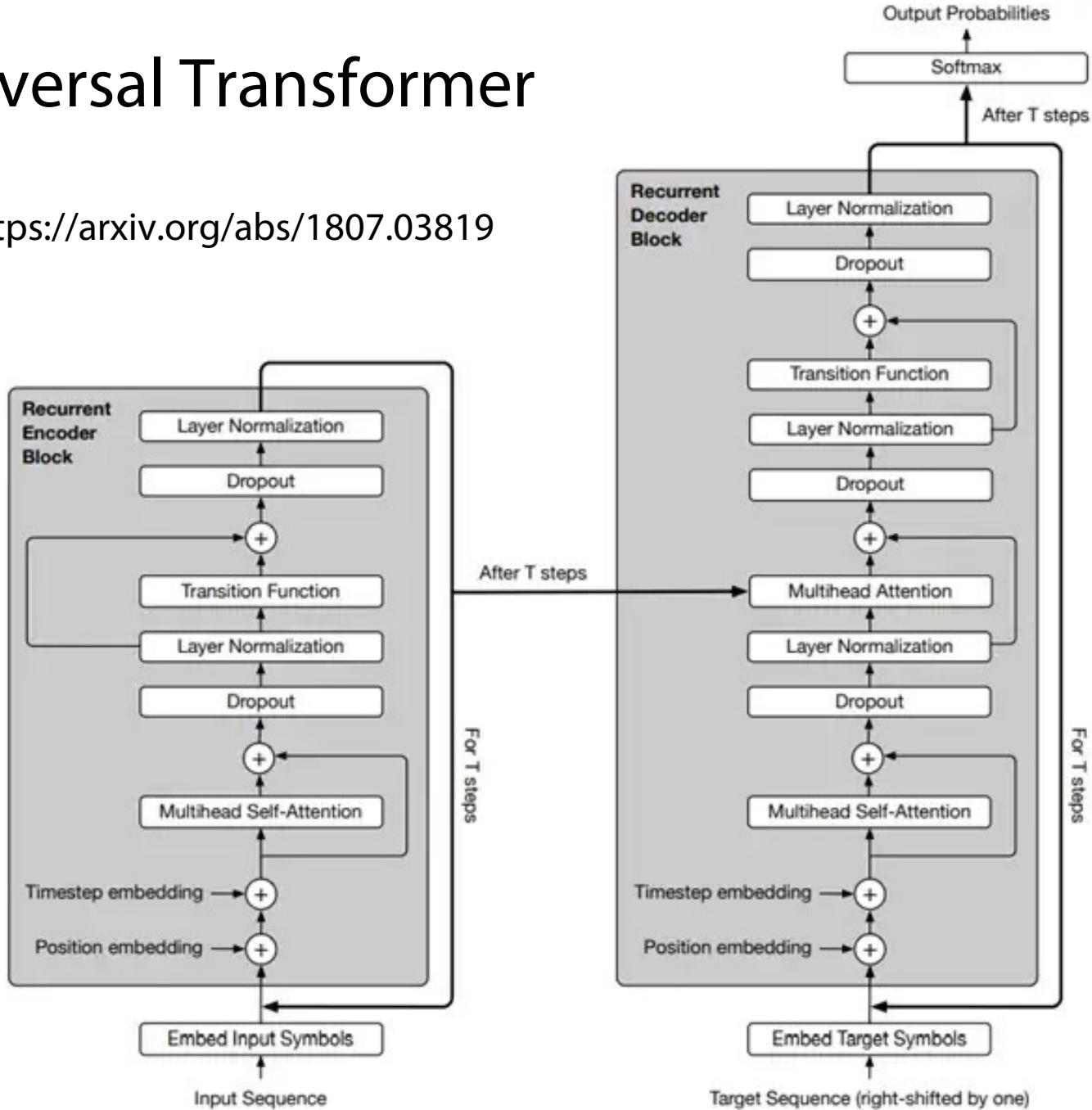
<https://ai.googleblog.com/2020/06/pegasus-state-of-art-model-for.html>

Summarization with attention-based AE + AR

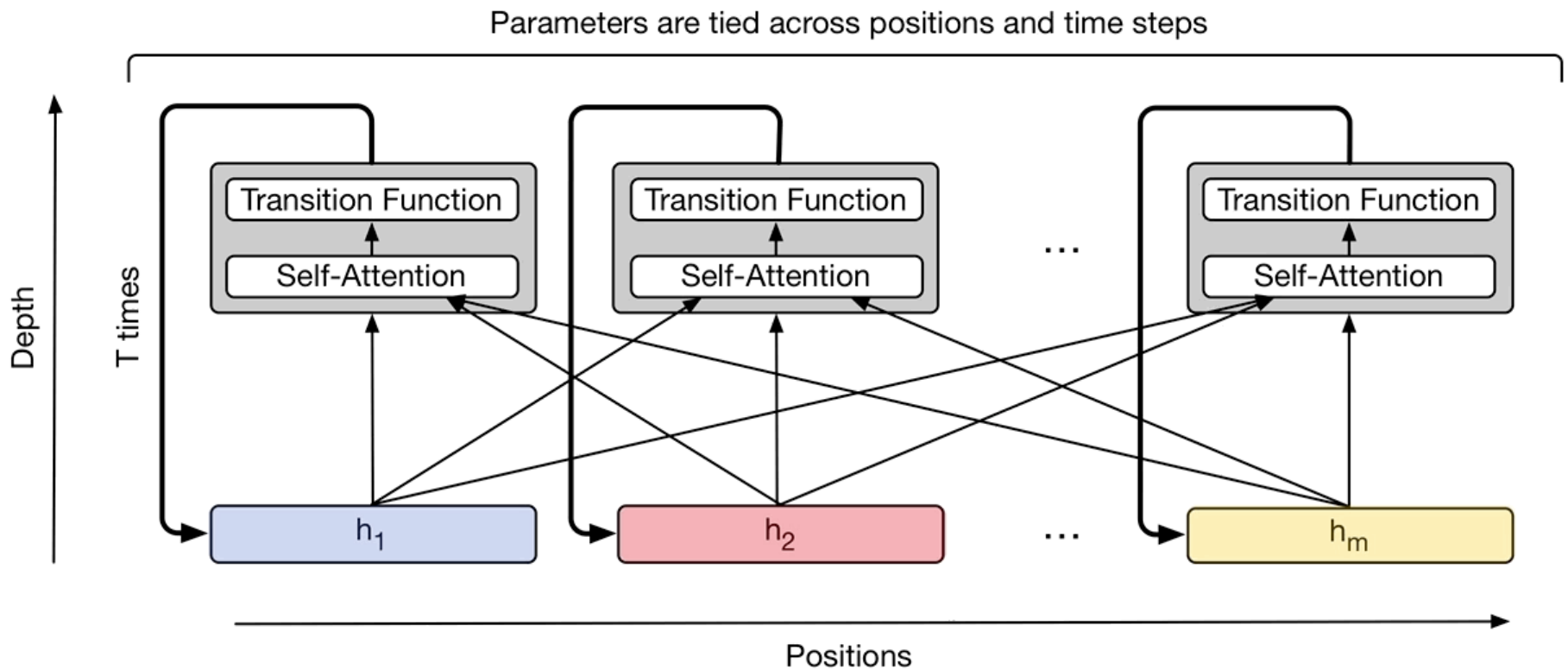


Universal Transformer

<https://arxiv.org/abs/1807.03819>



Universal Transformer



<https://ai.googleblog.com/2018/08/moving-beyond-translation-with.html>

Intelligent Agents

1d-CNNs LSTMs ELMo Transformers BERT GPT

Ralf Möller

Universität zu Lübeck

Institut für Informationssysteme



Interim Conclusion

- Transformers: Efficient, multi-modal data processors
 - Based on embedding technology
- Postneural AI: Finally, AI becomes effective
 - How much of our thoughts and conversation are just filling the gap reasoning?
 - How much of our thoughts and conversation are just next word prediction?
 - We just do not care as long as we have a real cool computing device
 - Example: GPT
- Recap the GPT family

GPT-1 (2018)

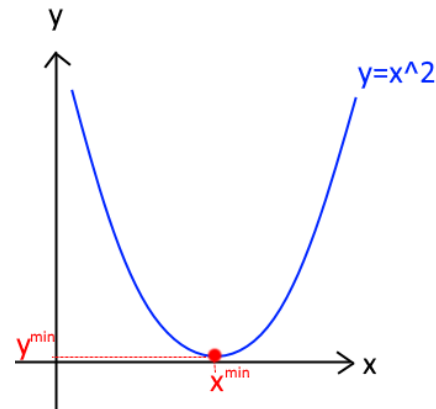
- Pre-cursor to BERT (2019)
- Similar architecture and training procedures
 - 117M parameters in GPT1 vs. 340M for BERT Large
- Pre-training: Maximize data likelihood as a product of conditional probabilities, trained on Books Corpus
 - Predict each token based on the k tokens (the “context”) that came before
- To be fine-tuned for each task while also retaining the generative objective
- Training and fine-tuning based on gradient descent

Gradient-Based function minimization (GD):

Function: $y = f(x)$ e.g. $y = x^2$

Minimize function:

- $\min f(x)$ – find smallest $f(x)$ value
- $\operatorname{argmin} f(x)$ – find x^{\min} s.t. $f(x^{\min})$ is smallest value



Could be a loss function

Gradient-Based function minimization (GD):

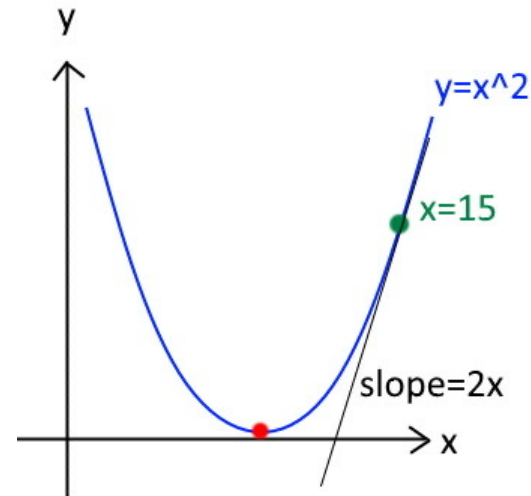
Function: $y = f(x)$ e.g. $y=x^2$

Minimize function:

- $\min f(x)$ – find smallest $f(x)$ value
- $\operatorname{argmin} f(x)$ – find x^{\min} s.t. $f(x^{\min})$ is smallest value

Gradient: $y' = f(x)' = 2x$ – slope or direction, where function grows

For simplification, gradient can be thought as -1 or +1



Transformers are composed of simple, differentiable functions: Gradient backpropagation yields informed search for a parametrization with loss minimization

Gradient-Based function minimization (GD):

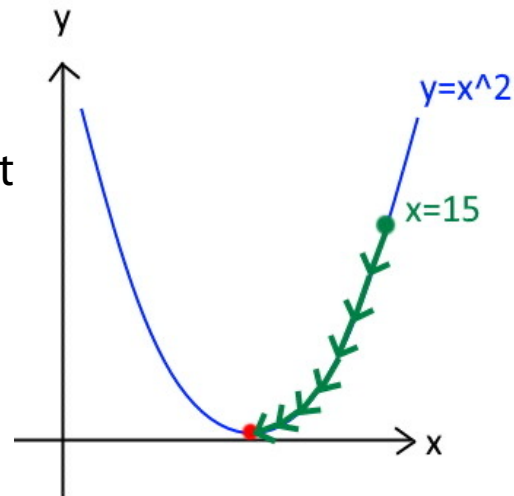
Function: $y = f(x)$ e.g. $y=x^2$

Minimize function:

- $\min f(x)$ – find smallest $f(x)$ value
- $\operatorname{argmin} f(x)$ – find x^{\min} s.t. $f(x^{\min})$ is smallest value

Gradient: $y' = f(x)' = 2x$ – slope or direction, where function grows

For simplification, gradient can be thought as -1 or +1



Gradient descent (naïve version):

1. $lr = 0.01$ # learning rate i.e. step size
2. $x = 15$ # start from random starting point
3. for i in $\operatorname{range}(200)$: # repeat until convergence
4. $\text{gradient} = 2*x$ # Compute gradient
5. $x = x - \text{gradient} * lr$ # Update parameter

Generalization: Vector input and output

Let

$$\mathbf{y} = \psi(\mathbf{x}), \quad (23)$$

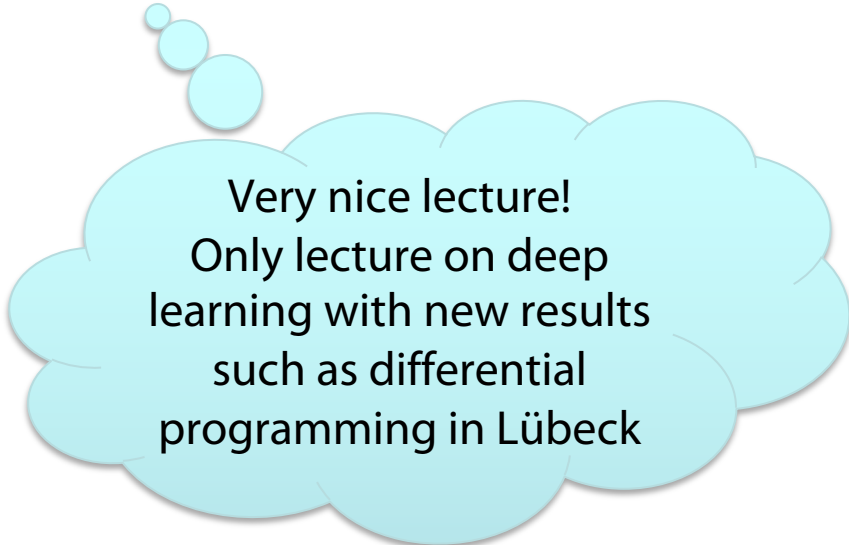
where \mathbf{y} is an m -element vector, and \mathbf{x} is an n -element vector. The symbol

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \dots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \dots & \frac{\partial y_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \dots & \frac{\partial y_m}{\partial x_n} \end{bmatrix} \quad (24)$$

will denote the $m \times n$ matrix of first-order partial derivatives of the transformation from \mathbf{x} to \mathbf{y} . Such a matrix is called the Jacobian matrix of the transformation $\psi(\cdot)$.

Probabilistic and Differential Programming

- Introduction
- Gradient descent
- Deep networks and Deep learning I: basic concepts
- Deep networks and Deep learning II: RNNs, Reservoir
- Embeddings: word2vec, knowledge graph embeddings, logic of cones
- Deep networks and Deep learning III: autoencoders (AEs), variational autoencoders (VAEs), generative adversarial networks (GANs)
- Automatic Differentiation of Programs
- Probabilistic Programming I
- Probabilistic Programming II
- Probabilistic Circuits I (Learning)
- Probabilistic Circuits II (Learning)
- Probabilistic Circuits III

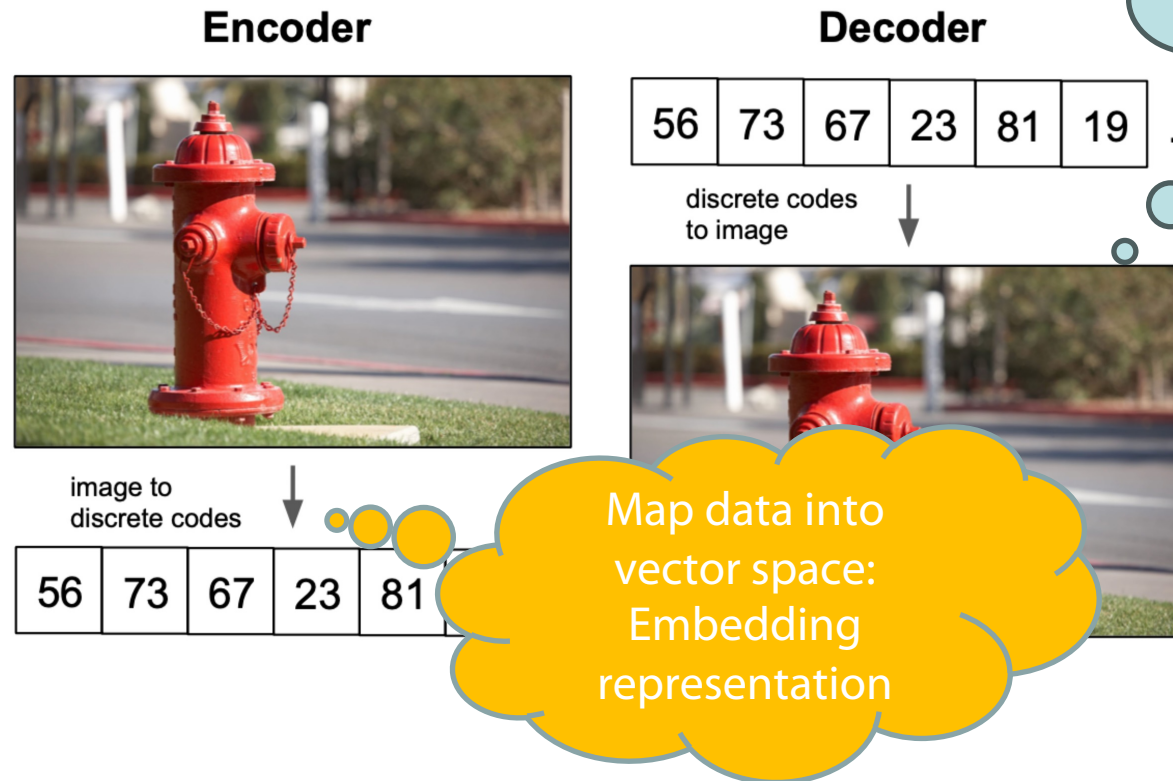


Very nice lecture!
Only lecture on deep
learning with new results
such as differential
programming in Lübeck

Gradient Descent

- Would you update with loss from one new input datum?
- Compute average gradient from training dataset containing many inputs
- How to handle multiple layers during backpropagation?
 - Gradient may vanish (stop the update?)
 - Gradient might also explode (depending on the loss function)
- Autoencoder to the rescue?

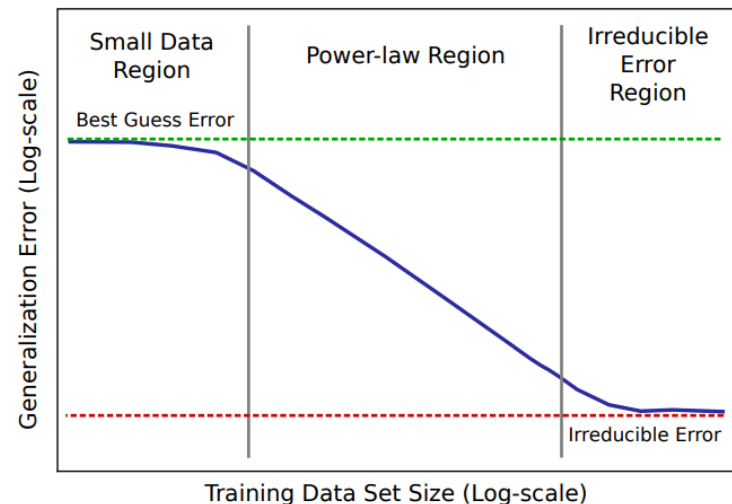
Autoencoder



Reconstruction-optimal encoding?

Stochastic Gradient Descent (SGD)

- GD: all the points in the training set are used to calculate the loss and its derivative
- SGD: only a single point or a small subset of points is used randomly for this purpose.
- SGD much faster and more suitable for large-scale datasets, ...
- ... but it is only an approximation of GD
- ... due to the introduction of more noise and variance in the gradient estimate



GPT-2: Multiple tasks supported

Language Models are Unsupervised Multitask Learners

Alec Radford ^{*1} Jeffrey Wu ^{*1} Rewon Child ¹ David Luan ¹ Dario Amodei ^{**1} Ilya Sutskever ^{**1}

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset of millions of webpages called WebText. When conditioned on a document plus questions, the answers generated by the language model reach 55 F1 on the CoQA dataset - matching or exceeding the performance of 3 out of 4 baseline systems without using the 127,000+ training examples. The capacity of the language model is essential

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

GPT-2

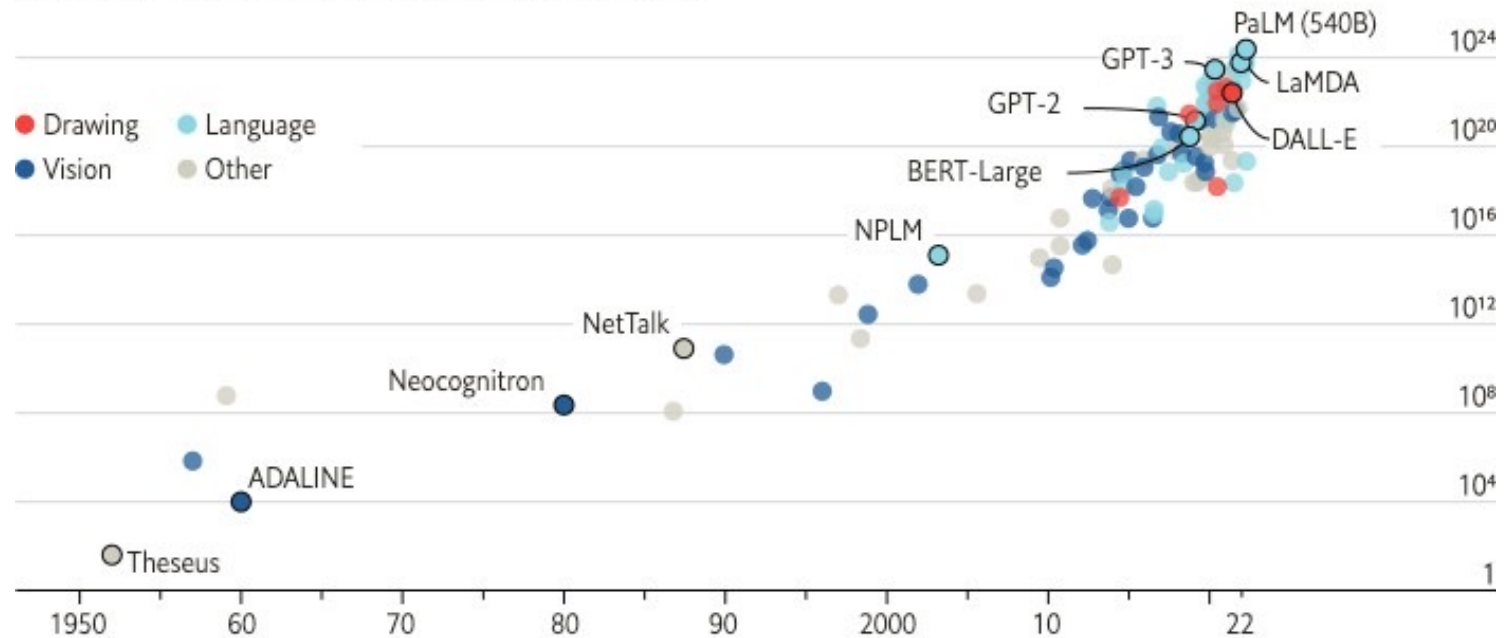
- A general systems should learn to model $P(\text{output}|\text{input}, \text{task})$
- The task can be specified in natural language, so language tasks can be framed as sequence-to-sequence text processing
 - Find task by prompt classification (could be done with BERT)
- Sequence-to-sequence: A problem formulated as receiving input in some modality and producing output in some modality (instead of e.g. predicting probability for labels in a specific task)
- GPT-2 is generatively trained on WebText data and not initially fine-tuned on anything else
- GPT-2 needs to be fine-tuned for handling specific contexts well

Larger and larger models: E.g. GPT-3

The blessings of scale

AI training runs, estimated computing resources used

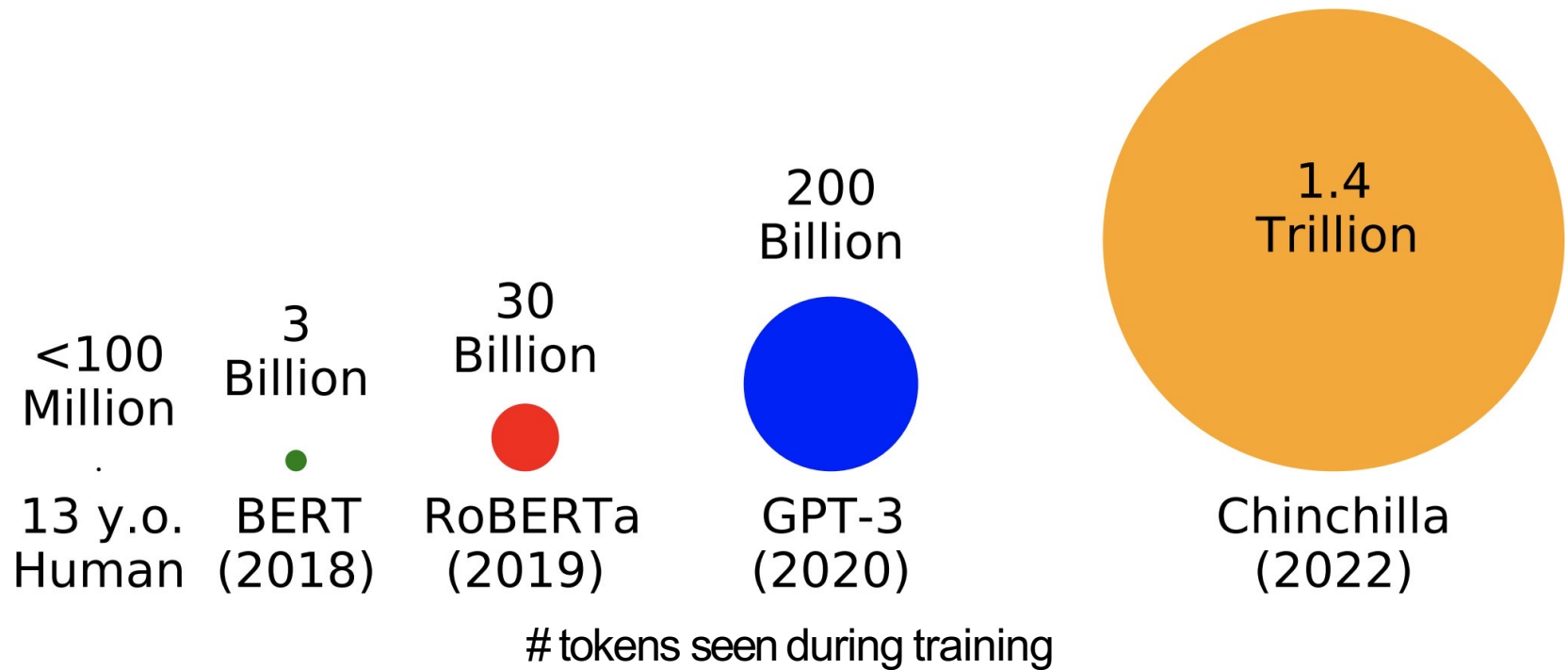
Floating-point operations, selected systems, by type, log scale



Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data

3 <https://www.economist.com/interactive/briefing/2022/06/11/huge-foundation-models-are-turbo-charging-ai-progress>

Trained on more and more data



<https://babylm.github.io/>

THE COST OF TRAINING NLP MODELS

A CONCISE OVERVIEW

Or Sharir
AI21 Labs
ors@ai21.com

Barak Peleg
AI21 Labs
barakp@ai21.com

Yoav Shoham
AI21 Labs
yoavs@ai21.com

April 2020

<http://arxiv.org/abs/2004.08900>

Costs: Not for the faint-hearted

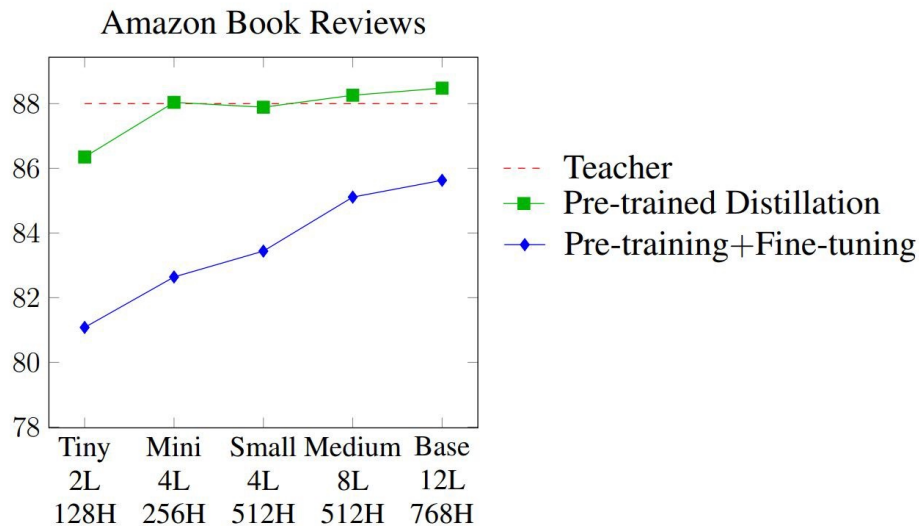
- \$2.5k - \$50k (110 million parameter model)
- \$10k - \$200k (340 million parameter model)
- \$80k - \$1.6m (1.5 billion parameter model)

Distillation to the rescue?

- A.k.a. model compression
- Idea has been around for a long time:
 - *Model Compression* (Bucila et al, 2006)
 - *Distilling the Knowledge in a Neural Network* (Hinton et al, 2015)
- Simple technique:
 - Train “Teacher”: Use SOTA pre-training + fine-tuning technique to train model with maximum accuracy
 - Label a large amount of unlabeled input examples with Teacher
 - Train “Student”: Much smaller model (e.g., 50x smaller) which is trained to mimic Teacher output
 - Student objective is typically Mean Square Error or Cross Entropy

Distillation

- Example distillation results
 - 50k labeled examples, 8M unlabeled examples



Well-Read Students Learn Better: On the Importance of Pre-training Compact Models
(Turc et al, 2020)

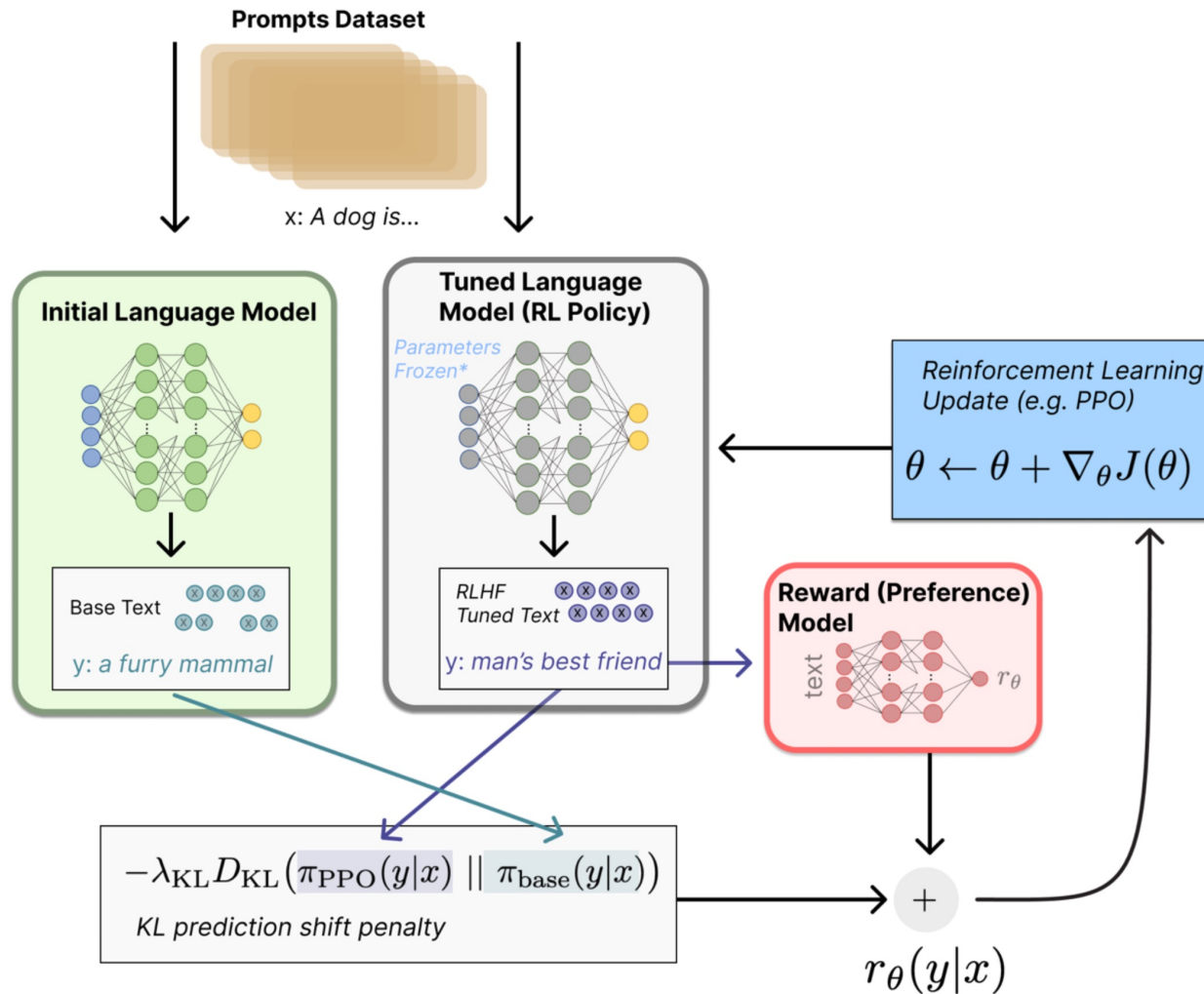
- Distillation works *much* better than pre-training + fine-tuning with smaller model

Why does distillation work so well?

A hypothesis:

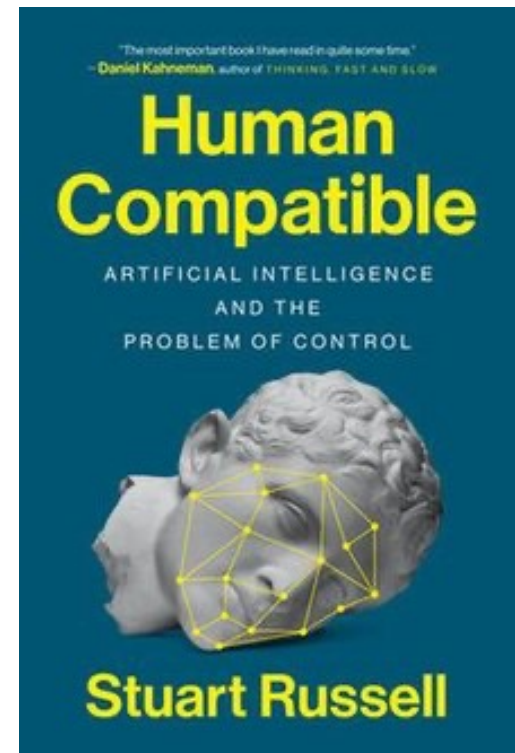
- Finetuning mostly just picks up and tweaks `_existing_` latent features
- This requires an oversized model, because only a subset of the features are useful for any given task
- Distillation allows the model to only focus on those features
- Supporting evidence: Simple self-distillation of a small model (e.g., distilling a smaller BERT model) doesn't work very well

GPT-3: Add RLHF with PPO (Recap)



Limitations of RL + Reward Modeling

- Human preferences are unreliable!
 - “Reward hacking” is a common problem in RL
 - Chatbots are rewarded to produce responses that seem authoritative and helpful, regardless of truth
 - This can result in making up facts
 - + hallucinations
- Models of human preferences are even more unreliable!
- There is a real concern of AI mis(alignment)!



GPT-3 Even more versatile w/ few-shot learning

Language Models are Few-Shot Learners

Tom B. Brown*

Benjamin Mann*

Nick Ryder*

Melanie Subbiah*

Jared Kaplan[†]

Prafulla Dhariwal

Arvind Neelakantan

Pranav Shyam

Girish Sastry

Amanda Askell

Sandhini Agarwal

Ariel Herbert-Voss

Gretchen Krueger

Tom Henighan

Rewon Child

Aditya Ramesh

Daniel M. Ziegler

Jeffrey Wu

Clemens Winter

Christopher Hesse

Mark Chen

Eric Sigler

Mateusz Litwin

Scott Gray

Benjamin Chess

Jack Clark

Christopher Berner

Sam McCandlish

Alec Radford

Ilya Sutskever

Dario Amodei

OpenAI



Example

Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // _____

LM

Circulation revenue has increased by 5% in Finland. // Finance

They defeated ... in the NFC Championship Game. // Sports

Apple ... development of in-house chips. // Tech

The company anticipated its operating profit to improve. // _____

LM

- In-context learning is competitive with models trained with much more labeled data and is state-of-the-art on LAMBADA (commonsense sentence completion) and TriviaQA (question answering)
- Other examples: Writing code from natural language descriptions, helping with app design mockups, and generalizing spreadsheet functions

GPT-3 "In-Context" Learning

Zero-Shot

1 Translate English to French: ← task description
2 cheese => ← prompt

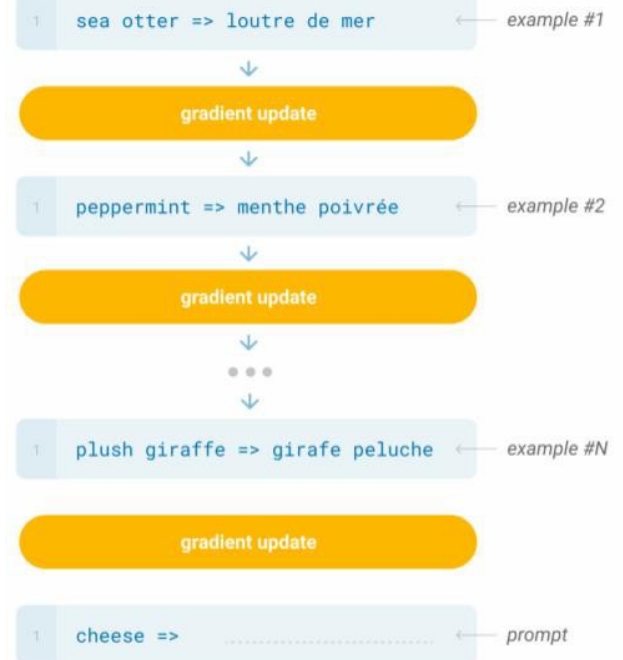
One-Shot

1 Translate English to French: ← task description
2 sea otter => loutre de mer ← example
3 cheese => ← prompt

Few-Shot

1 Translate English to French: ← task description
2 sea otter => loutre de mer ← examples
3 peppermint => menthe poivrée
4 plush girafe => girafe peluche
5 cheese => ← prompt

Traditional Fine-Tuning



In-context learning: Analysis

- In-context learning describes a different paradigm of “learning”
- where the model is fed input normally as if it were a black box,
- and the input to the model describes a new task with some possible examples
- while the resulting output of the model reflects that new task as if the model had “learned”

- How does this new paradigm compare to “pretrain + finetune”?

TriviaQA

Question

Miami Beach in Florida borders which ocean?

What was the occupation of Lovely Rita according to the song by the Beatles

Who was Poopdeck Pappys most famous son?

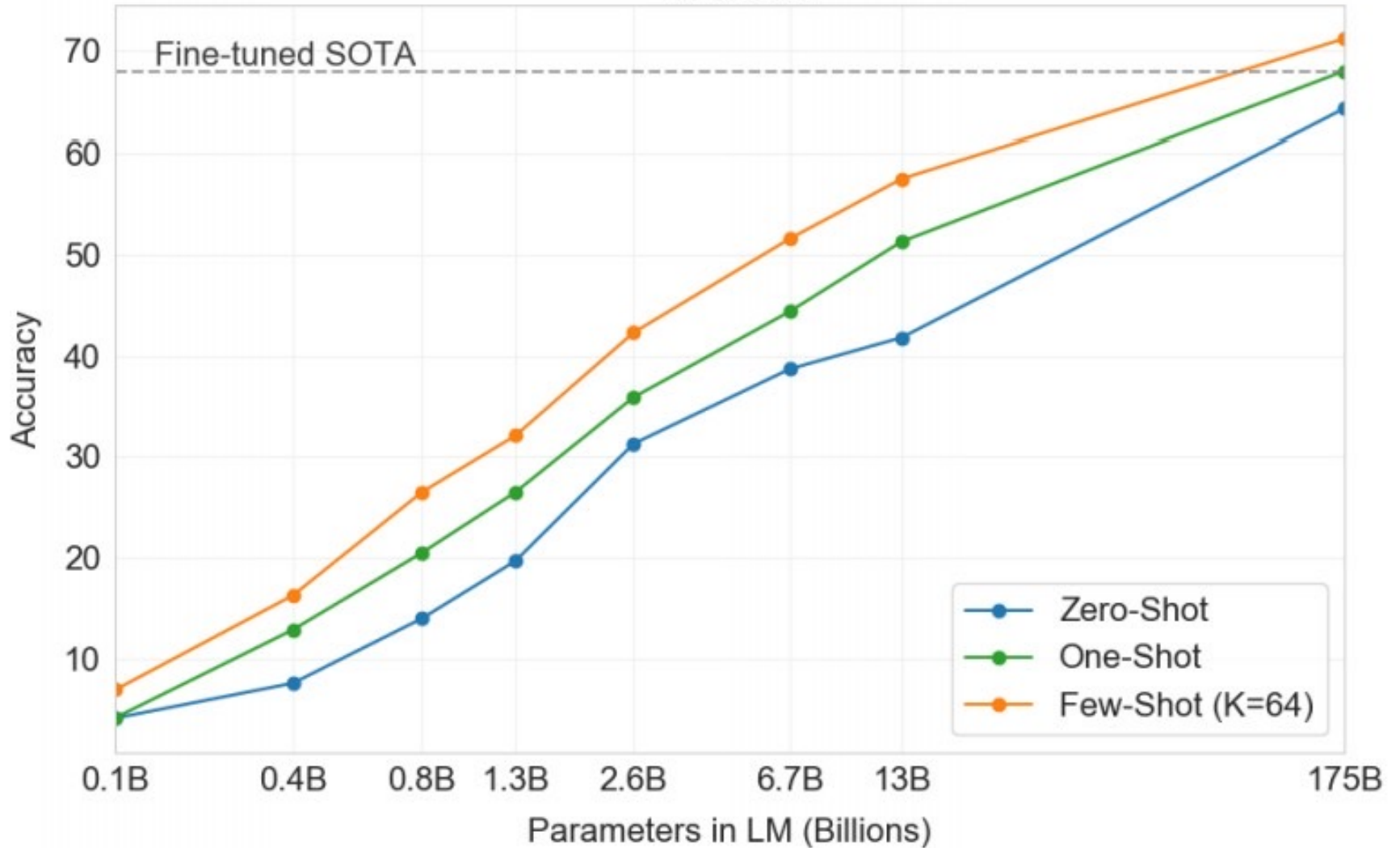
The Nazi regime was Germany's Third Reich; which was the first Reich?

At which English racecourse did two horses collapse and die in the parade ring due to electrocution, in February 2011?

Which type of hat takes its name from an 1894 novel by George Du Maurier where the title character has the surname O'Ferrall ?

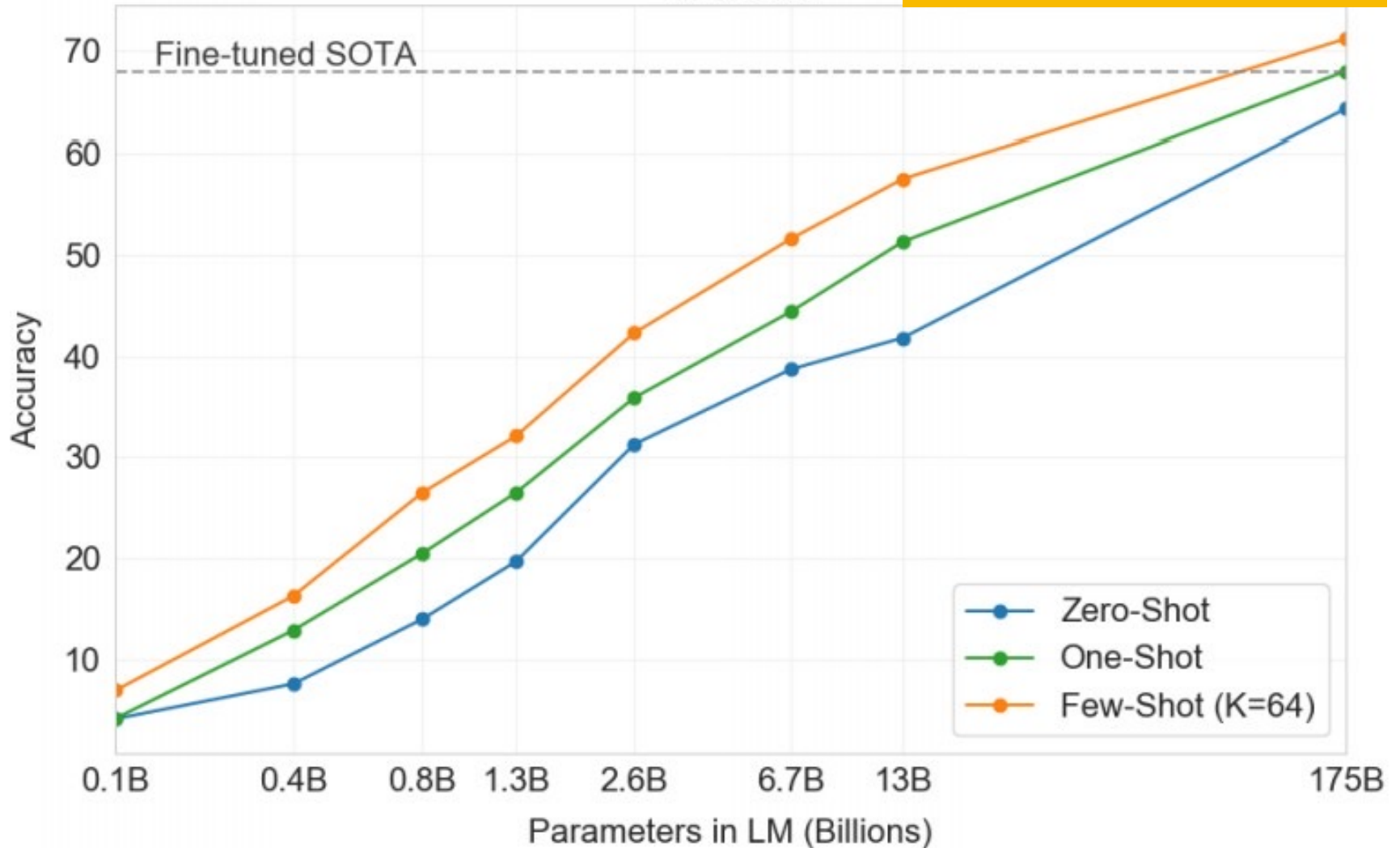
What was the Elephant Man's real name?

TriviaQA



TriviaQA

What does this mean?



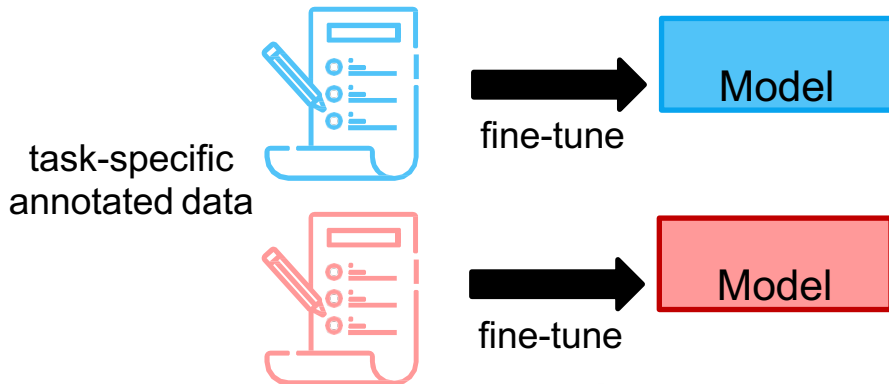
Pretraining and Fine-tuning

Powerful Pre-Trained Model – GPT 3

Pre-Training & Fine-Tuning

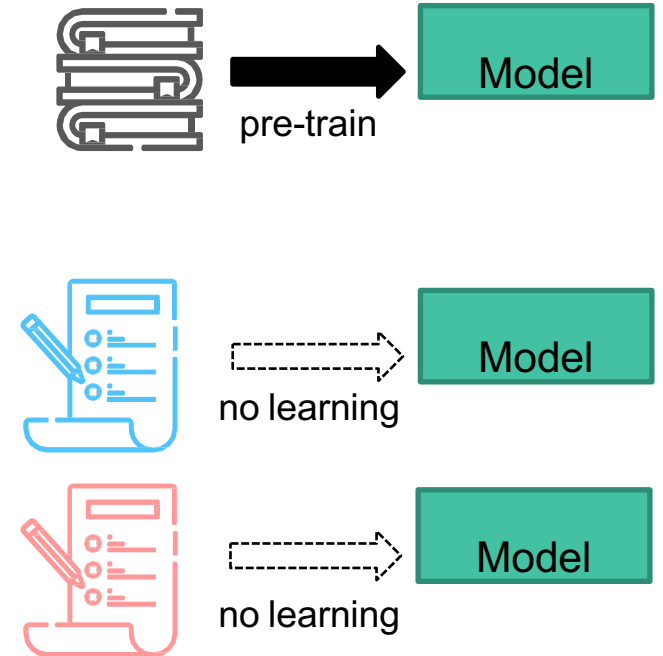


Instruct GPT



Need to store new params

Pre-Training & In-Context Learning

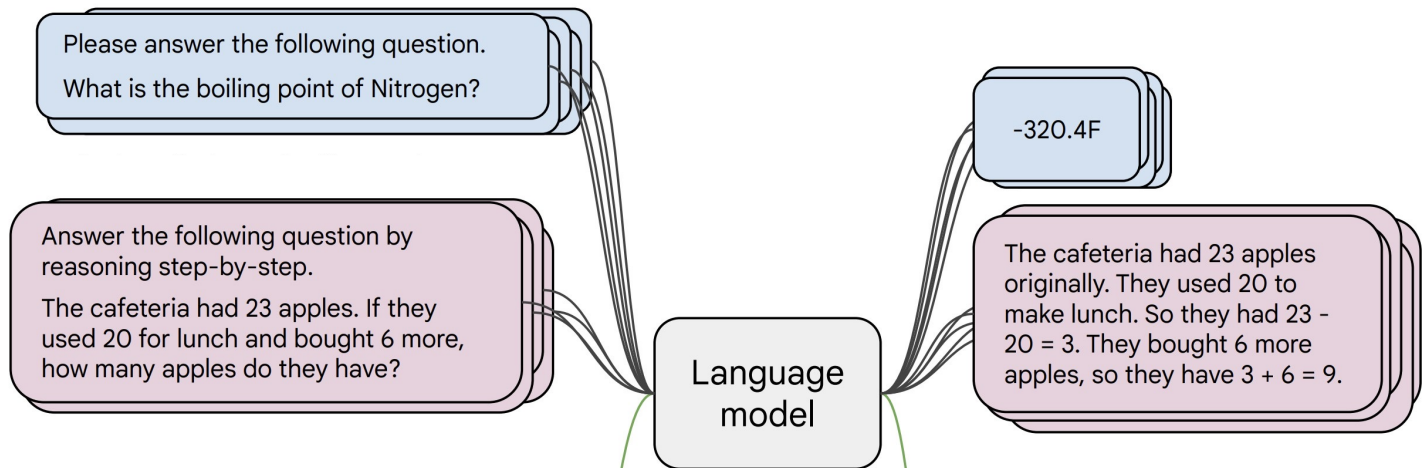


Understanding context learning

- An Explanation of In-context Learning as Implicit Bayesian Inference
 - <https://arxiv.org/abs/2111.02080>
- Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?
 - <https://arxiv.org/abs/2202.12837>
- Can the context be given by principals when giving task descriptions that are passed to agents?
- Can in-context learning help us with dialogues?

Instruction finetuning

- **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM



- Evaluate on **unseen tasks**



Summarization


Prompt

> Summarise the impact of gall ink on manuscript production as reported in the journal "manuscript cultures" edited by the Center of the Studies of Manuscript Cultures in Hamburg.


Perplexity.AI

Sources

Chapter 4 Material Studies of Historic Inks:...

 brill · 1

[PDF] manuscript cultures - CSMC -...

 uni-hamburg · 2

Iron Gall Ink – NYC Department of Records ...

 archives · 3



View 2 more

The impact of iron-gall ink on manuscript production, as reported in the journal "Manuscript Cultures" edited by the Center for the Study of Manuscript Cultures in Hamburg, is significant. Iron-gall inks, which dominated the black to brown palette of writing materials in manuscripts from the Middle Ages to the 19th century, have been a fundamental part of written cultural heritage. However, these inks pose a formidable challenge for the preservation of manuscripts due to their corrosive nature. The ink's corrosive effects endanger many Western historical manuscripts, as it can lead to the degradation and loss of the cellulose-based support, particularly the paper, on which it is used. Over time, the ink can fade, create a haloing effect around letters, or even corrode the paper support, leaving areas of loss resembling burn marks. While the ink's indelibility and rich dark tone were desirable traits for scribes and artists, its damaging effects on paper are well-documented. The transition from carbon ink to iron-gall ink, which was extremely popular in the medieval Middle East and Europe, has been the focus of investigative work, combining a search of extant written records with experimental determinations of the ink. The journal "Manuscript Cultures" and related publications from the Center for the Study of Manuscript Cultures contribute to the interdisciplinary study of manuscript cultures, including material analysis, and provide new insights into the impact of iron-gall ink on manuscript production.

Too Long; Didn't Read – TL;DR:

Prompt

TL;DR: The idea that agency can be attributed to inanimate objects, not just humans, and that these can be considered non-human actors has become popular in anthropology, art history and the social sciences since being introduced in the late 1990s by British social anthropologist Alfred Gell and then expanded by French sociologist Bruno Latour.1 The concept here – the agency of objects – has been applied to diverse topics concerned with material culture studies since then. It has not resulted in a unified theory being created, though. Rather, various theoretical approaches have been pursued in parallel. These vary greatly in terms of their theoretical frameworks, but they also differ as to what kinds of non-human actors are considered. The range includes everything from everyday items to works of art, buildings and even the social structures and networks connected with them. Hardly anyone has thought about the agency of manuscripts yet, though, or applied the concept systematically to research on manuscripts or written artefacts in general. Partly on the grounds of this disparity between existing theoretical approaches to the subject of agency, when dealing with the agency of written artefacts, it generally proves advantageous not to take a top-down approach based on the idea that just one concept of agency is valid.2 Rather, the subject should preferably be approached with a more open mind, i.e. exploratively when examining manuscripts, and then drawing on appropriate theoretical models. This open-mindedness is also reflected in the contributions to this volume, which draw on highly differing approaches to agency. In view of this diversity, the fundamental understanding of agency formulated

Prompt text from: “Agency: How Manuscripts Affect and Create Social Realities” by Michael Kohs and Sabine Kienitz | Hamburg

Sources

- The agency of things: how spaces and...
tandfonline · 1
- About the Agency of Things, of Objects and...
uni-heidelberg · 2
- artefacts in theory: anthropology and...
jstor · 3
- View 2 more

Answer

The concept of the agency of objects, introduced by social anthropologist Alfred Gell and expanded by sociologist Bruno Latour, has gained popularity in anthropology, art history, and the social sciences. It suggests that inanimate objects can have agency and be considered non-human actors, impacting human behavior and social structures. This idea has been applied to diverse topics in material culture studies, including everyday items, works of art, buildings, and social structures. However, there is no unified theory of the agency of objects, and various theoretical approaches have been pursued in parallel, with a wide range of non-human actors being considered. While the agency of manuscripts has not been widely explored, it is suggested that a more open-minded and explorative approach should be taken when examining manuscripts, drawing on appropriate theoretical models. The agency of manuscripts can shape social and cultural practices, but due to the diversity of theoretical approaches to agency, it is advantageous not to take a top-down approach based on just one concept of agency ¹ ² ⁵.

Share Rewrite



Related

- what is the concept of agency in anthropology and the social sciences +
- how has the concept of agency been applied to material culture studies +
- what are some examples of non-human actors that have been considered in the concept of agency +

Follow-up questions as prompt inspiration

Center of the Studies of Manuscript Cultures in Hamburg.

Sources

Centre for the Study of Manuscript Cultures... uni-hamburg · 1

Centre for the Studies of Manuscript Cultures... gerit · 2

manuscript cultures uni-hamburg · 3

View 2 more



Answer

The Center for the Study of Manuscript Cultures (CSMC) in Hamburg is a research institution affiliated with the University of Hamburg. It is dedicated to the interdisciplinary study of manuscript cultures, encompassing research from various disciplines such as art history, codicology, epigraphy, history, material analysis, palaeography, and philology. The center publishes the "Manuscript Cultures" (MC) journal, which encourages comparative approaches without regional, linguistic, temporal, or other limitations on the objects studied. The journal is peer-reviewed and openly accessible, contributing to a larger historical and systematic survey of the role of written artifacts in cultural studies. The CSMC's work aims to provide a new foundation for ongoing discussions in cultural studies and advance scholarship in the field of manuscript cultures.

Share Rewrite     

Related

- what is the mission of the centre for the study of manuscript cultures in hamburg +
- what are the research areas of the centre for the study of manuscript cultures in hamburg +
- what are the ongoing projects of the centre for the study of manuscript cultures in hamburg +

 Ask follow-up... Copilot 



Chain-of-thought prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

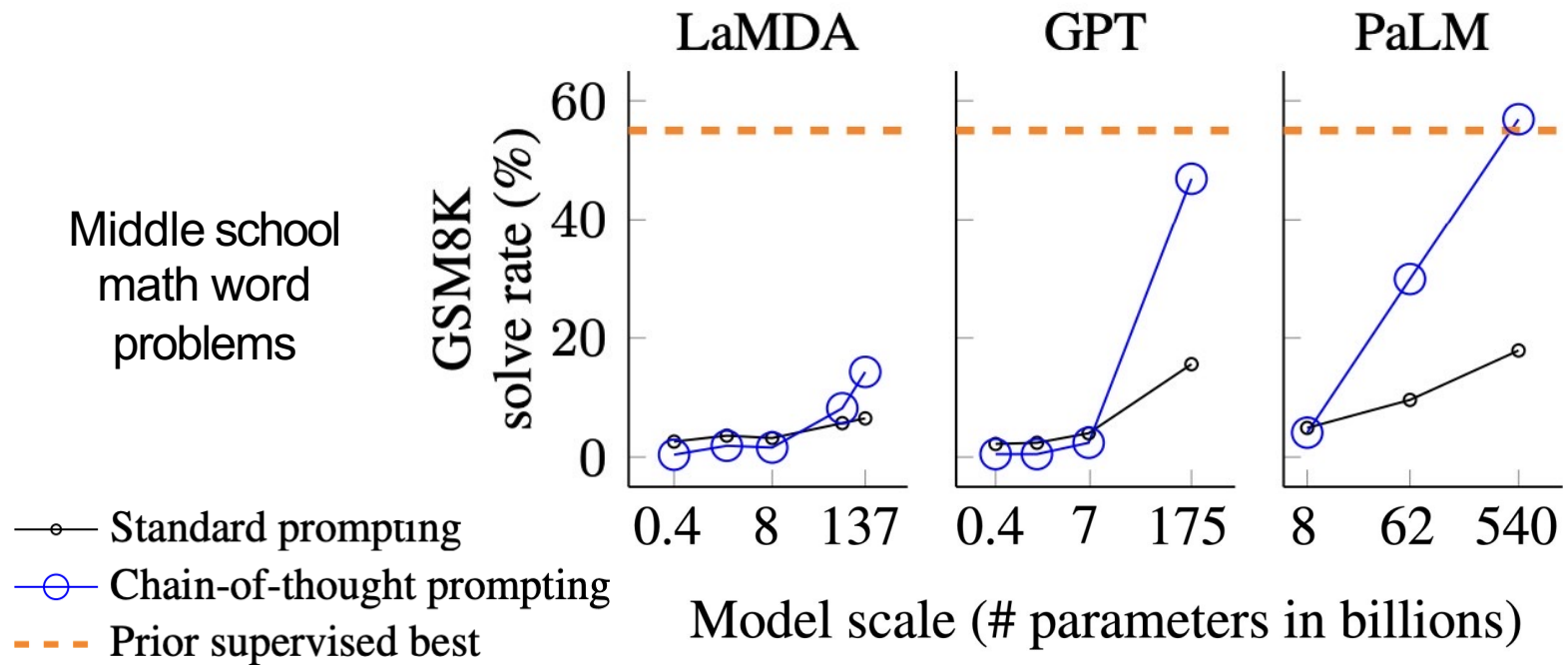
A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

Think step by step and explain the calculation step by step.

[\[Wei et al., 2022; also see Nye et al., 2021\]](#)

Chain-of-thought prompting

Emergent property of model scale



[\[Wei et al., 2022; also see Nye et al., 2021\]](#)

Chain-of-thought prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

Do we even need examples of reasoning?
Can we just ask the model to reason through things?

[\[Wei et al., 2022; also see Nye et al., 2021\]](#)

Zero-shot chain-of-thought prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.** There are 16 balls in total. Half of the balls are golf balls. That means there are 8 golf balls. Half of the golf balls are blue. That means there are 4 blue golf balls. ✓

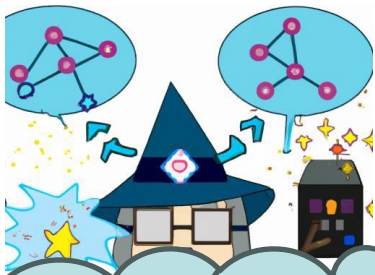
[[Kojima et al., 2022](#)]

Zero-shot chain-of-thought prompting

	MultiArith	GSM8K
Zero-Shot	17.7	10.4
Few-Shot (2 samples)	33.7	15.6
Few-Shot (8 samples)	33.8	15.6
Zero-Shot-CoT	Greatly outperforms zero-shot → 78.7	40.7
Few-Shot-CoT (2 samples)	84.8	41.3
Few-Shot-CoT (4 samples : First) (*1)	89.2	-
Few-Shot-CoT (4 samples : Second) (*1)	90.5	-
Few-Shot-CoT (8 samples)	Manual CoT still better → 93.0	48.7

[[Kojima et al., 2022](#)]

Zero-shot chain-of-thought prompting

No.	Category	Zero-shot CoT Trigger Prompt	Accuracy
1	LM-Designed	Let's work this out in a step by step way to be sure we have the right answer.	82.0
2	 Human-Designed	Let's think step by step. (*1)	78.7
3		First, (*2)	77.3
4		Let's think about this logically.	74.5
5		Let's solve this problem by splitting it into steps. (*3)	72.2
6		Let's be realistic and think step by step.	70.8
7		Let's think like a detective step by step.	70.3
8		Let's think	57.5
			Before we dive into the answer,
		The answer is after the proof.	45.7
		(Zero-shot)	17.7

New dark art of prompt engineering?

[\[Zhou et al., 2022; Kojima et al., 2022\]](#)

ChatGPT: Add conversation data




- Interactive, conversational model
- Part of GPT-3.5 family
 - fine-tuned mostly on programming code
- ChatGPT
 - is a sibling model to InstructGPT
 - ChatGPT is similar but not identical
 - slight differences in the data collection setup
 - a fine-tuned version of GPT-3.5 that's essentially a general-purpose chatbot
- Dialogue format of ChatGPT makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests

Language models as multitask assistants?

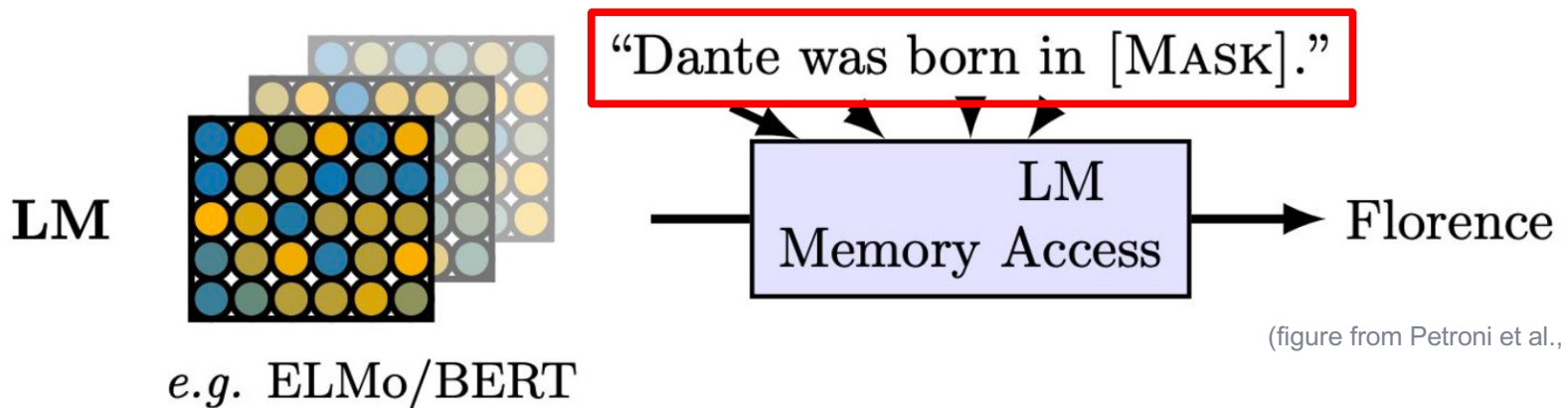
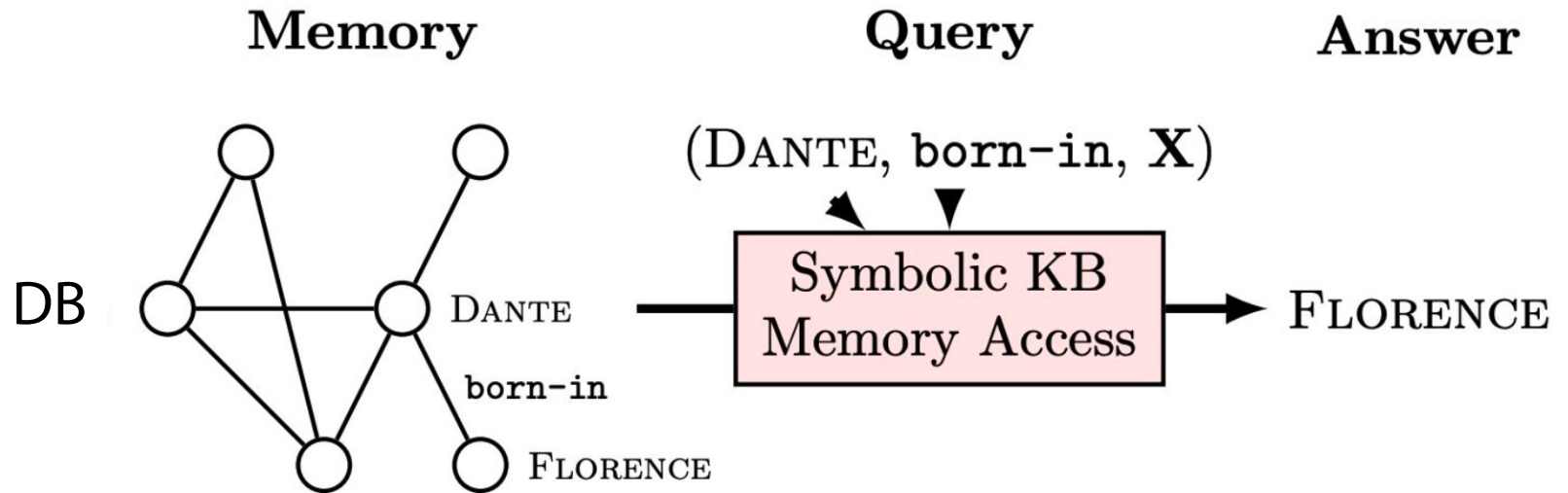
We've finally (mostly) answered how we get from this

Stanford University is located in _____

to *this*

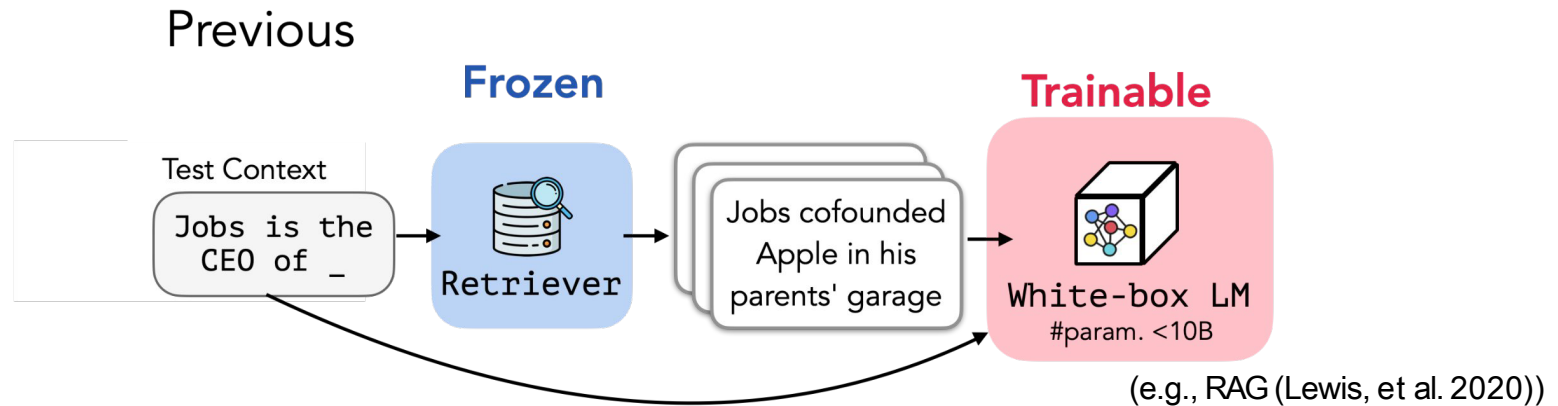
ChatGPT		
 Examples	 Capabilities	 Limitations
"Explain quantum computing in simple terms"	Remembers what user said earlier in the conversation	May occasionally generate incorrect information
"Got any creative ideas for a 10 year old's birthday?"	Allows user to provide follow-up corrections	May occasionally produce harmful instructions or biased content
"How do I make an HTTP request in Javascript?"	Trained to decline inappropriate requests	Limited knowledge of world and events after 2021

Data in LM's parameter space?

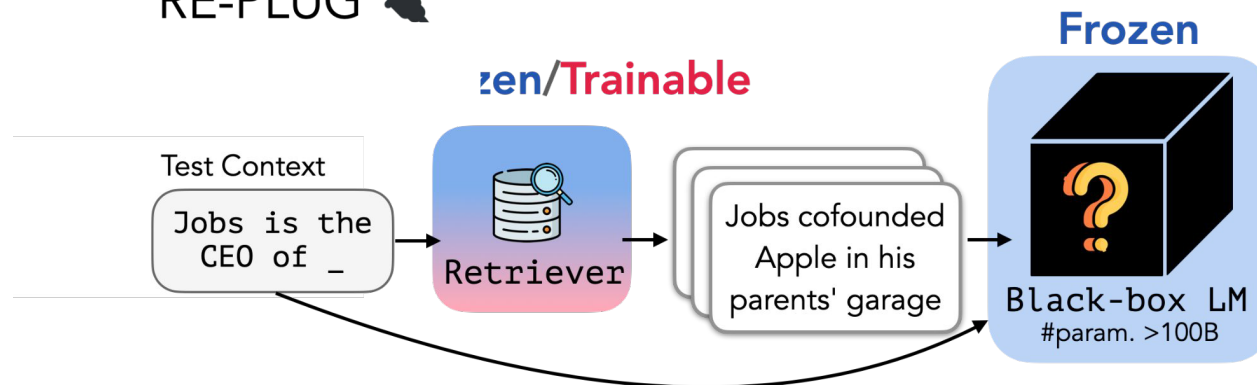


(figure from Petroni et al., 2019)

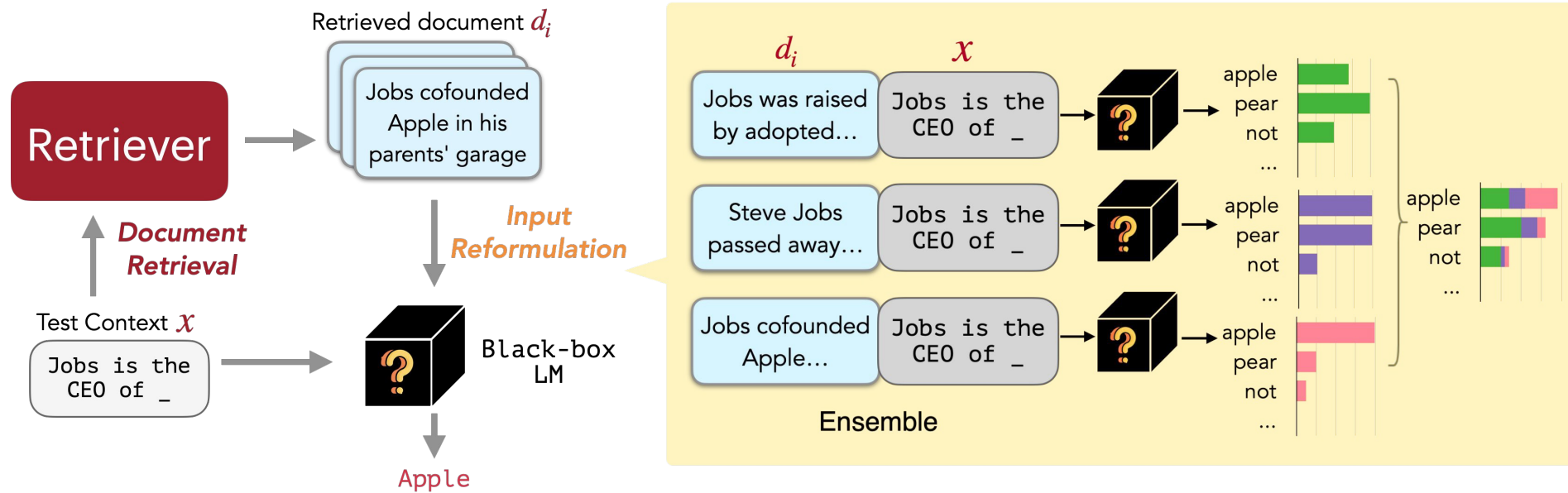
Retrieval-Augmented Generation (RAG)



RE-PLUG 



Information Fusion: REPLUG



<https://arxiv.org/abs/2301.12652>

Back to Agents

- Why is few-shot learning important?
- Agents can use pretrained model
- Few-shot learning for specifying dedicated tasks!

- From language models to general intelligence?

Language models as world models?

Language models may do rudimentary modeling of *agents*, *beliefs*, and *actions*:

Pat watches a demonstration of a bowling ball and a leaf being dropped at the same time in a vacuum chamber. Pat, who is a physicist, predicts that the bowling ball and the leaf will fall at the same rate.

Changing the last sentence of the prompt, we get:

... Pat, who has never seen this demonstration before, predicts that the bowling ball will fall to the ground first. This is incorrect. In a vacuum chamber, there is no air

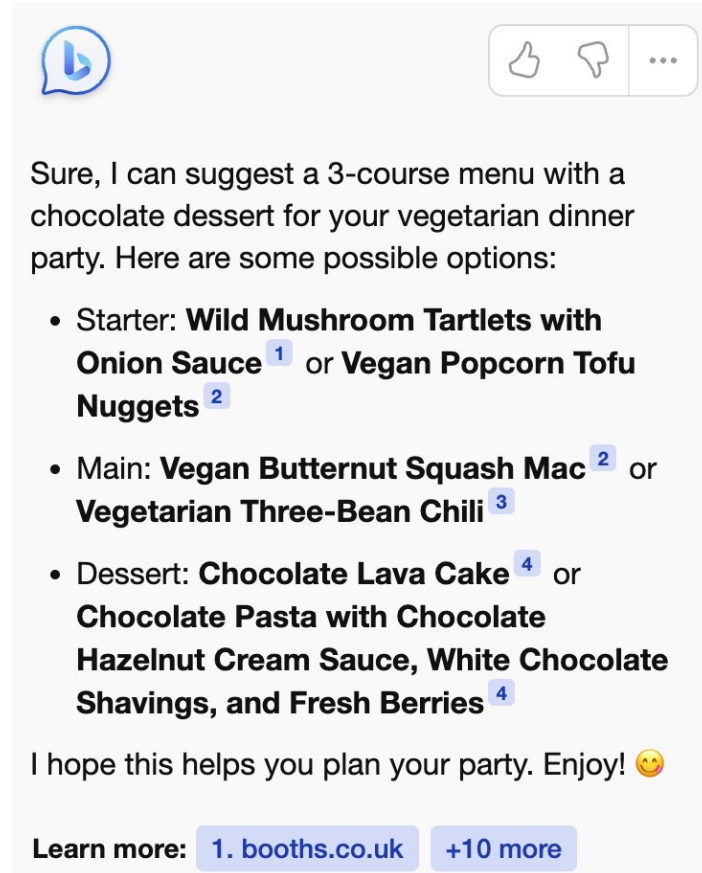
Language Models as Agent Models [[Andreas, 2022](#)]

Language models as multitask assistants?

I need to throw a dinner party for 6 people who are vegetarian. Can you suggest a 3-course menu with a chocolate dessert?

[[Microsoft Bing](#)]

(Also see OpenAI's ChatGPT, Google's Bard, Anthropic's Claude)



The screenshot shows a chat interface with a blue speech bubble icon on the left and thumbs up/down and a menu icon on the right. The text of the chat response is as follows:

Sure, I can suggest a 3-course menu with a chocolate dessert for your vegetarian dinner party. Here are some possible options:

- Starter: **Wild Mushroom Tartlets with Onion Sauce**¹ or **Vegan Popcorn Tofu Nuggets**²
- Main: **Vegan Butternut Squash Mac**² or **Vegetarian Three-Bean Chili**³
- Dessert: **Chocolate Lava Cake**⁴ or **Chocolate Pasta with Chocolate Hazelnut Cream Sauce, White Chocolate Shavings, and Fresh Berries**⁴

I hope this helps you plan your party. Enjoy! 😊

Learn more: [1. booths.co.uk](#) [+10 more](#)

agentgpt.reworkd.ai

AgentGPT Beta

Interested in automating businesses with AI Agents? Apply here >

AgentGPT-3.5 (2 / 5 runs)

Create an agent by adding a name / goal, and hitting deploy! Try our examples below!

ResearchGPT 📄

Create a comprehensive report of the Nike company

TravelGPT 🌴

Plan a detailed trip to Hawaii.

StudyGPT 📖

Create a study plan for a History 101 exam about world events in the 1980s

Special case of AutoGPT – more abstract

Name	CurriculumPlanner	Tools ⚙️
★ Goal	Develop a curriculum for a study programme on artificial intelligence and philosophy.	

▶️ || ⏏

Reworkd

CurriculumPlanner

Pages

- 📄 Templates New
- ❓ Help
- ⚙️ Settings
- 👤 Manage account

🔄
🐦
💬
📺

👉 **Subscribe**

Ralf Möller
moeller@uni-luebeck.de

★ **Embarking on a new goal** Develop a curriculum for a study programme on artificial intelligence and philosophy.

✦ Starting task: Identify the key topics and concepts in artificial intelligence and philosophy that should be covered in the study programme.

✦ **Task Added:** Research and gather relevant resources, textbooks, and articles on artificial intelligence and philosophy.

✦ **Task Added:** Identify the key topics and concepts in artificial intelligence and philosophy that should be covered in the study programme.

✦ **Task Added:** Design the structure and sequence of the curriculum, including the order in which topics will be taught and the duration of each session.

✦ **Task Added:** Create lesson plans, assignments, and assessments to ensure effective teaching and learning.

🔄 Generating response...

✔ **Finished:** Identify the key topics and concepts in artificial intelligence and philosophy that should be covered in the study programme.

✔ **Executing:** Identify the key topics and concepts in artificial intelligence and philosophy that should be covered in the study programme. 📄

To develop a curriculum for a study program on artificial intelligence and philosophy, it is essential to identify the key topics and concepts that should be covered. Here are some key topics and concepts that should be included in the study program:

1. **Introduction to Artificial Intelligence and Philosophy:** Provide an overview of artificial intelligence and philosophy, their history, and how they intersect. This will help students understand the context and importance of studying these fields together.
2. **Ethical Considerations in Artificial Intelligence:** Explore the ethical implications of artificial intelligence, such as privacy, bias, and accountability. Discuss the ethical frameworks and dilemmas that arise in AI development and application.
3. **Logic and Reasoning:** Cover the fundamental principles of logic and reasoning in both artificial intelligence and philosophy. This includes propositional and predicate logic, deductive and inductive reasoning, and their applications in AI systems and philosophical arguments.
4. **Philosophy of Mind:** Examine philosophical theories and debates surrounding the nature of mind, consciousness, and perception. Discuss how AI systems attempt to simulate or replicate human cognition and the implications of these efforts.
5. **Machine Learning and Neural Networks:** Introduce machine learning algorithms, including supervised and unsupervised learning, reinforcement learning, and neural networks. Explore their philosophical foundations and implications for AI research and development.

⬅ Back

🗑 Delete

➦ Share

- 🏠 Home
- 📄 Templates New
- 🔍 Help
- ⚙ Settings
- 👤 Manage account
- 🗨
- 🐦
- 💬
- 🌐

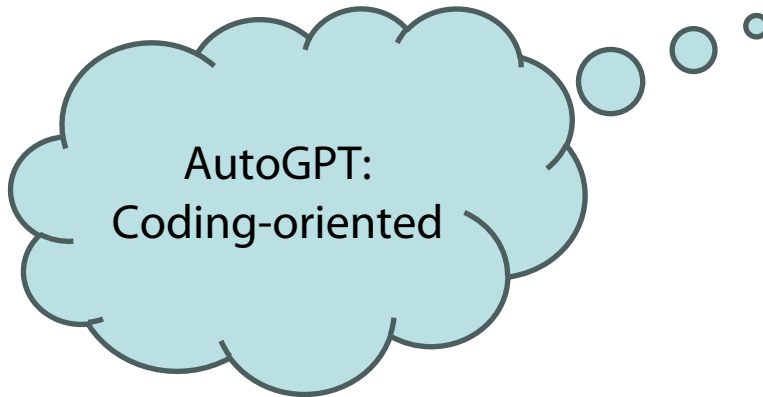
👉 **Subscribe**

 **Ralf Möller**
moeller@uni-luebeck.de



Abilities

- Abilities are the gears and levers that enable the agent to interact with tasks at hand
- Example: Web page fetcher



```
import requests

@ability(
    name="fetch_webpage",
    description="Retrieve the content of a webpage",
    parameters=[
        {
            "name": "url",
            "description": "Webpage URL",
            "type": "string",
            "required": True,
        }
    ],
    output_type="string",
)
async def fetch_webpage(agent, task_id: str, url: str) -> str:
    response = requests.get(url)
    return response.text
```


OpenAI: GPT Builder

- GPTs
- Custom versions of ChatGPT created by OpenAI users
 - All you have to do is tell the GPT builder, in plain English, what you want to create, and the builder will take it from there.

<https://help.openai.com/en/articles/8554397-creating-a-gpt>

<https://help.openai.com/en/articles/8770868-gpt-builder>

A Survey on Large Language Model based Autonomous Agents

**Lei Wang, Chen Ma*, Xueyang Feng*, Zeyu Zhang, Hao Yang, Jingsen Zhang,
Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei,
Ji-Rong Wen**

Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China

