

Ralf Möller, Sylvia Melzer

1d-CNNs LSTMs ELMo Transformers BERT GPT

Acknowledgements

- Some slides are based on
 - 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
 - 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)
 - Slides have been modified (All errors are mine)

Convolution

Input image



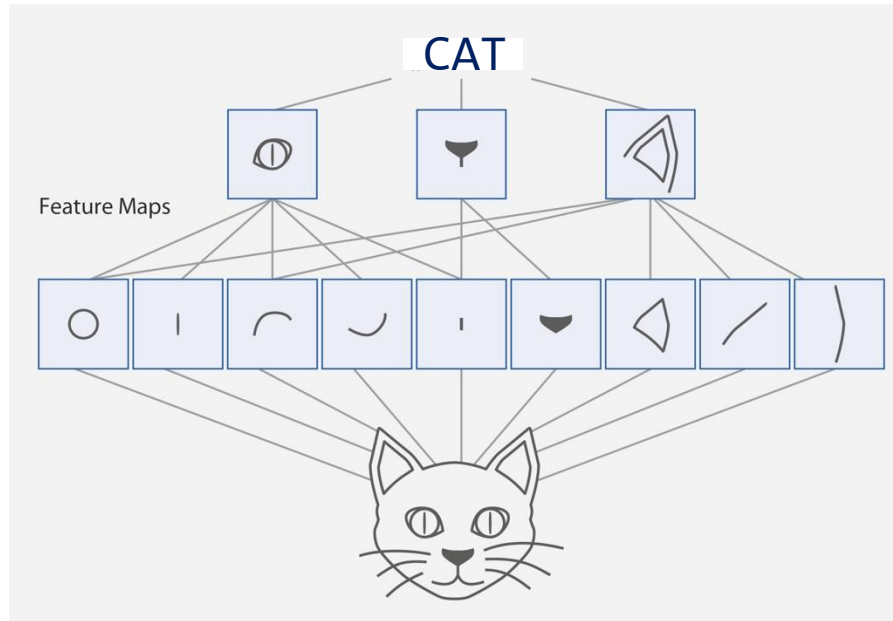
Convolution
Kernel

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

Feature map

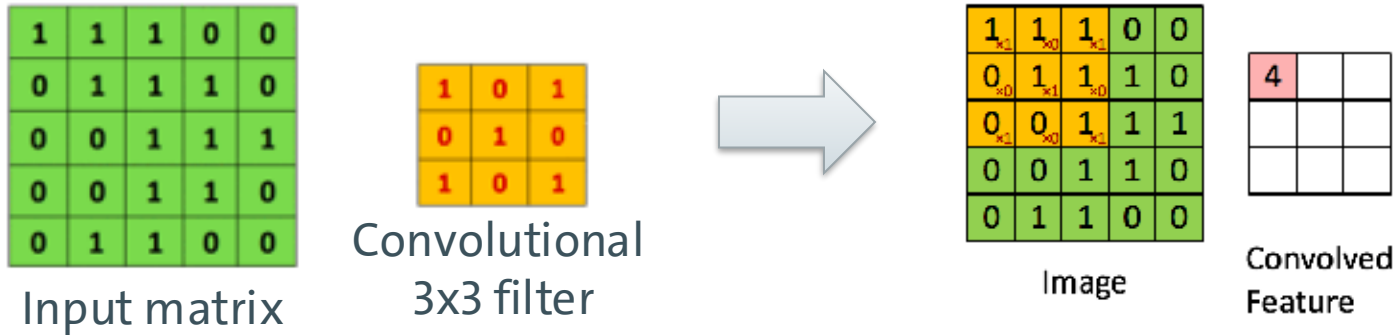


Feature Maps



<https://www.heise.de/ratgeber/Neuronale-Netz-einfach-erklart-6343697.html?seite=6>

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

ConvNets (CNs)

Feature Map

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

Max-Pooling

Main ConvNet idea for text:

Compute vectors for n-grams and group them afterwards

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

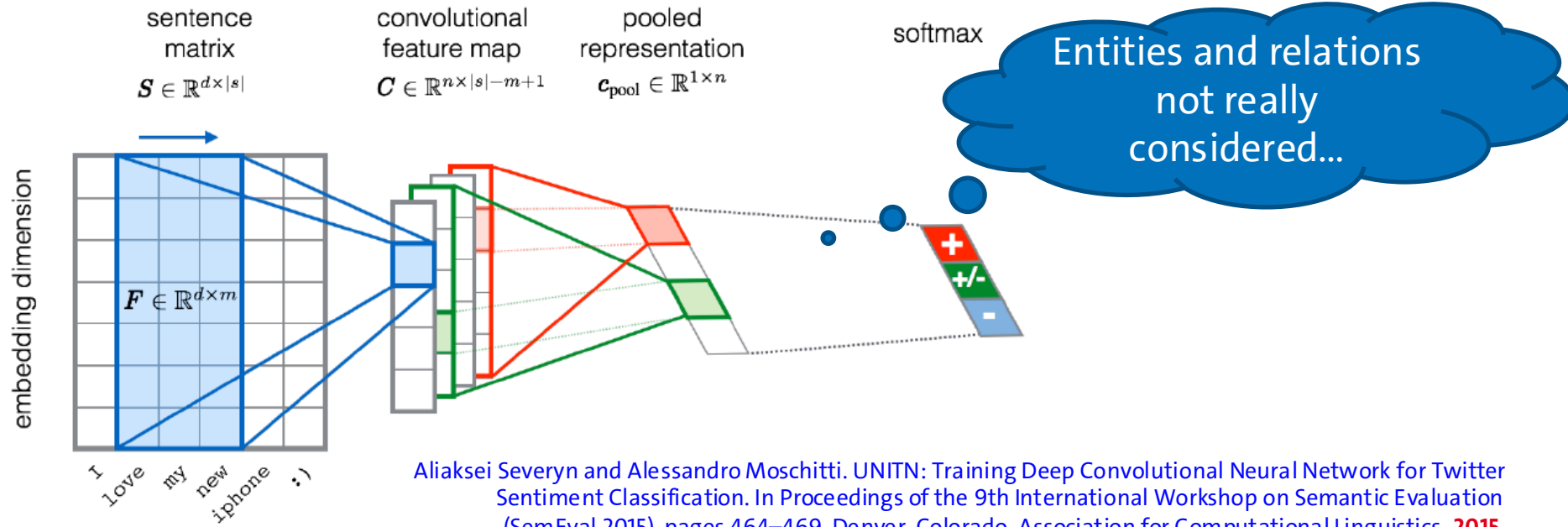
1d-CNNs for text

- 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

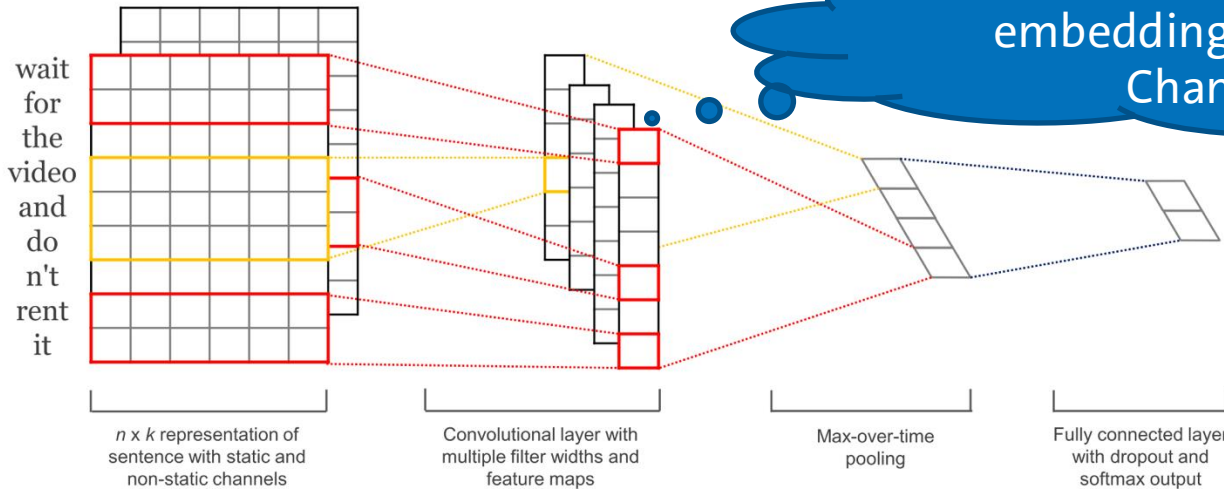
- CNNs (w/ ReLU and maxpool) can be used for classifying (parts of) the text

CNNs for sentiment analysis



Aliaksei Severyn and Alessandro Moschitti. UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification. In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 464–469, Denver, Colorado. Association for Computational Linguistics. 2015.

CNNs for sentence/text classification

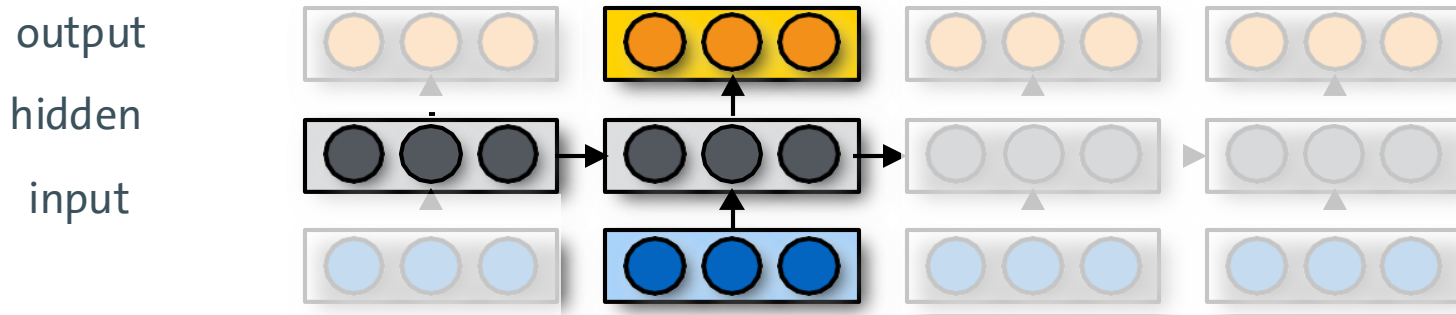


sliding over 3, 4 or 5 words at a time

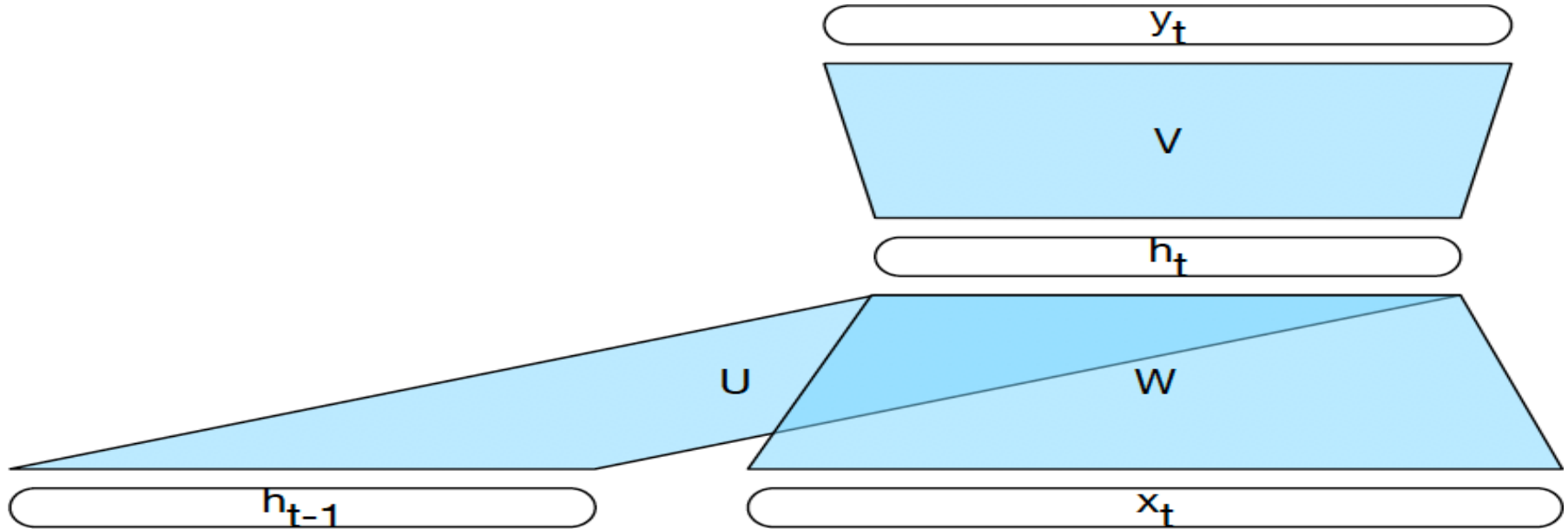
Yoon Kim. [Convolutional Neural Networks for Sentence Classification](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics. **2014**.

Recurrent Networks (RN) – Or: Copying the Pattern

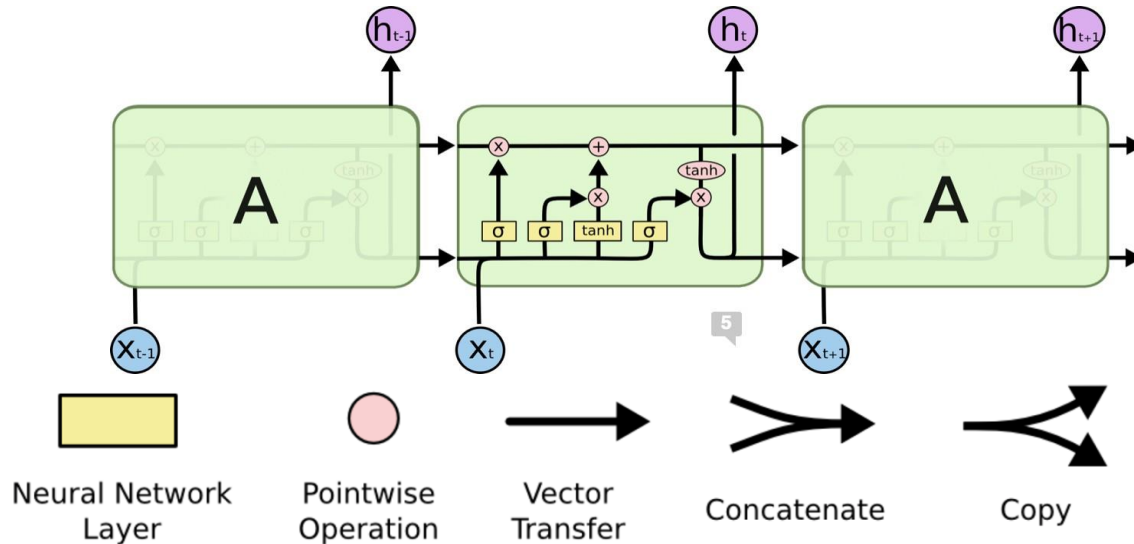
- Basic computational network copied per time slice
- Input: previous hidden state, output: next hidden state



Computing the Hidden State



Long Short Term Memory Networks (LSTMs)



Sepp Hochreiter, Jürgen Schmidhuber:
Long Short-Term Memory.
 In: *Neural Computation*. 9, 1997, S. 1735,
[doi:10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735)

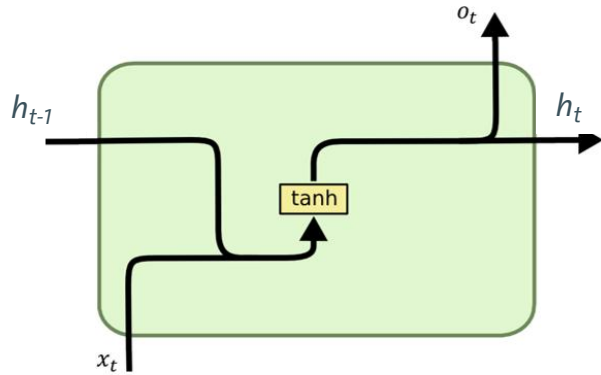
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Repetitive Variants: LSTMs, GRUs

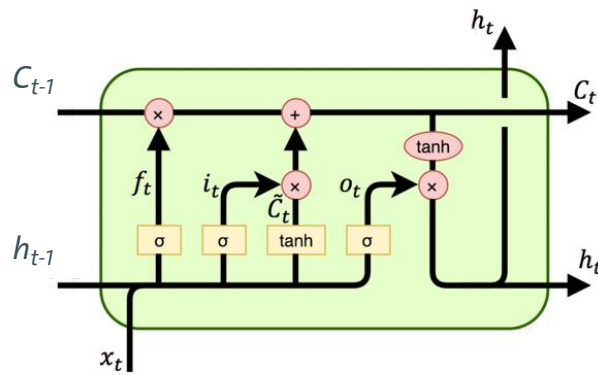
- **Long Short Term Memory** networks (LSTMs) are RNNs with a more complex architecture to combine the last hidden state with the current input.
- **Gated Recurrent Units** (GRUs) are a simplification of LSTMs
- Both contain “**gates**” to control how much of the input or past hidden state to forget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

Repetitive Variants: LSTMs, GRUs

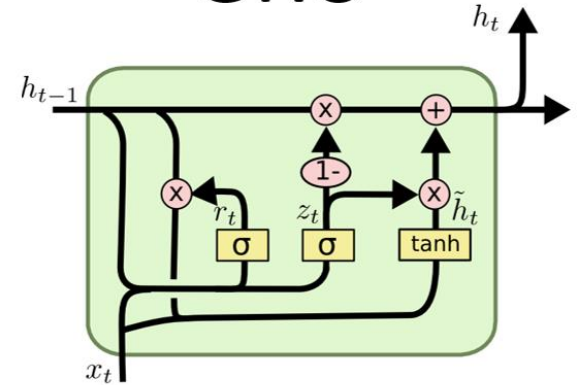
RN



LSTM

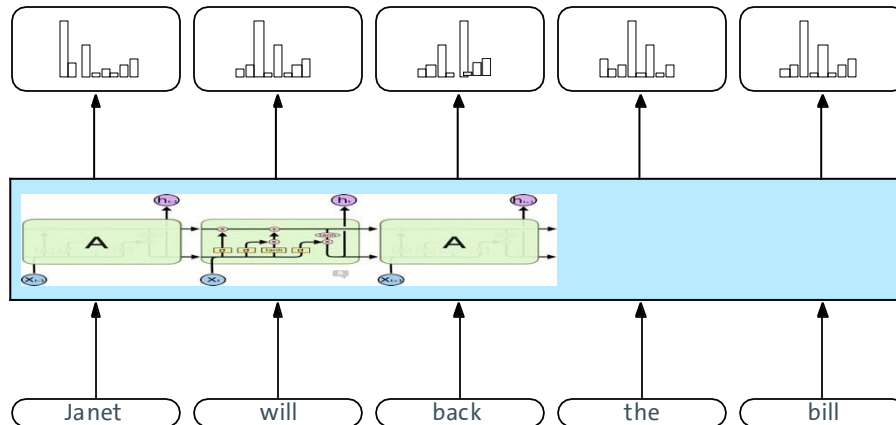


GRU

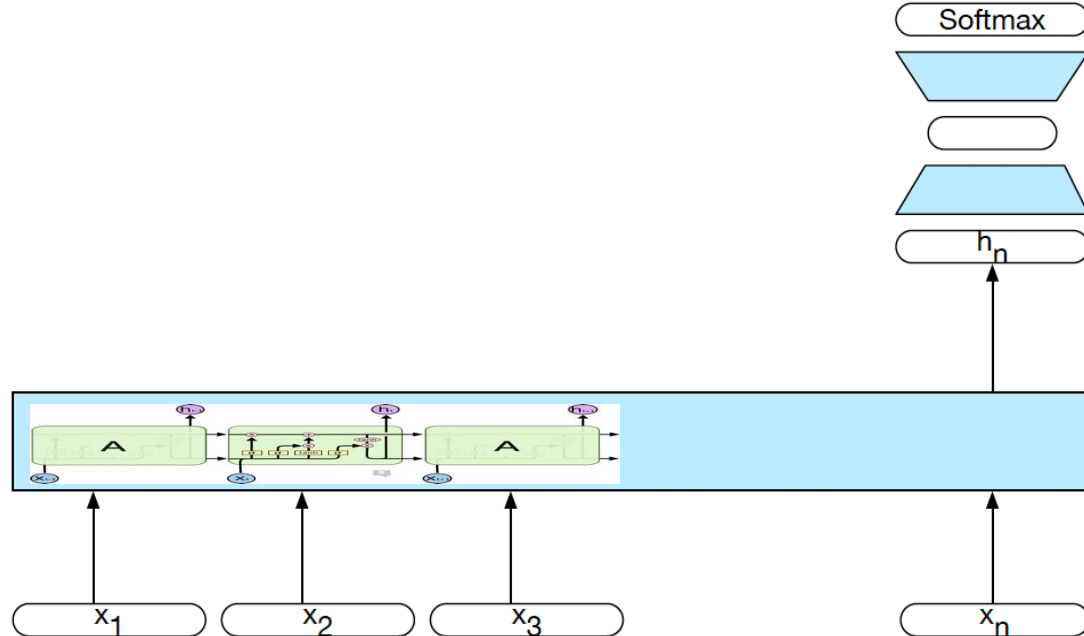


Basic RNs for Sequence Labeling

- Each time step has a distribution over output classes

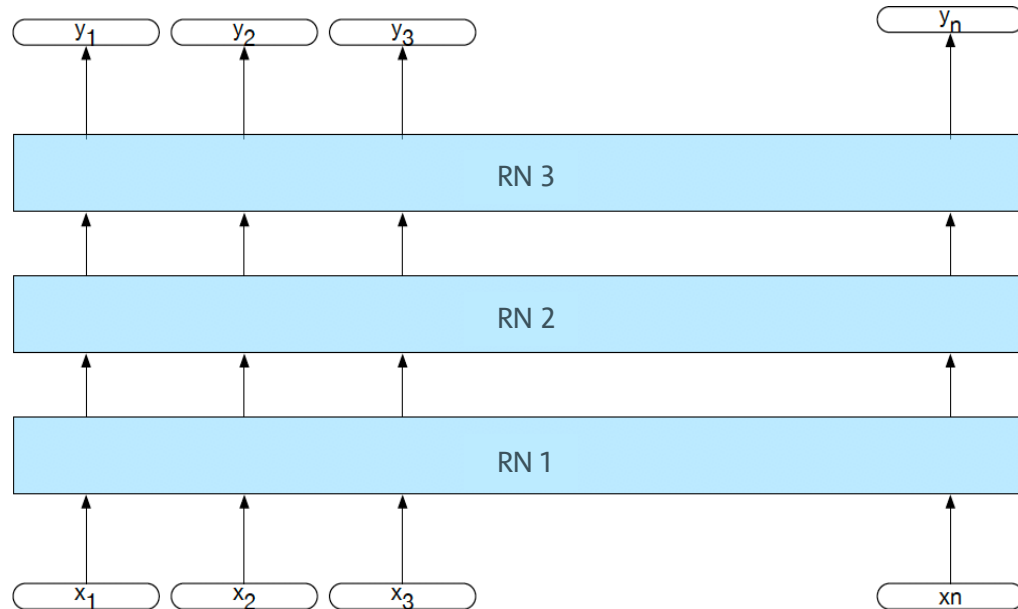


RNs for Sequence Classification

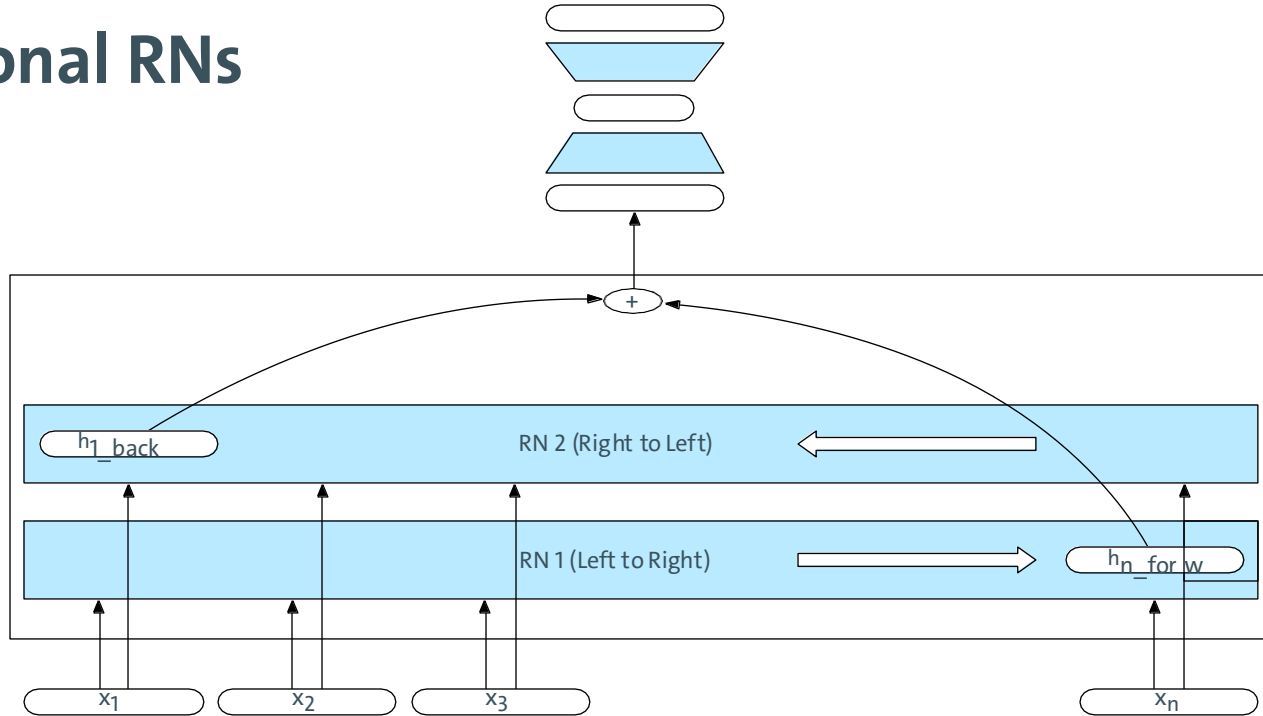


Stacked RNs

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



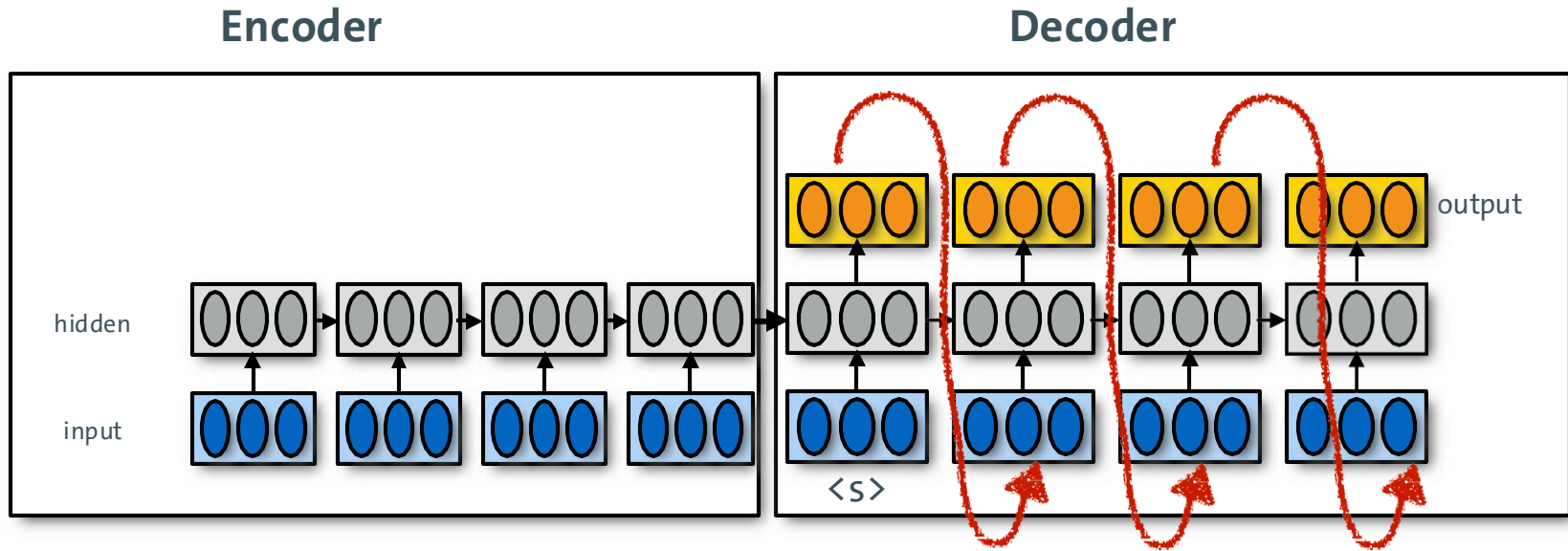
Bidirectional RNs



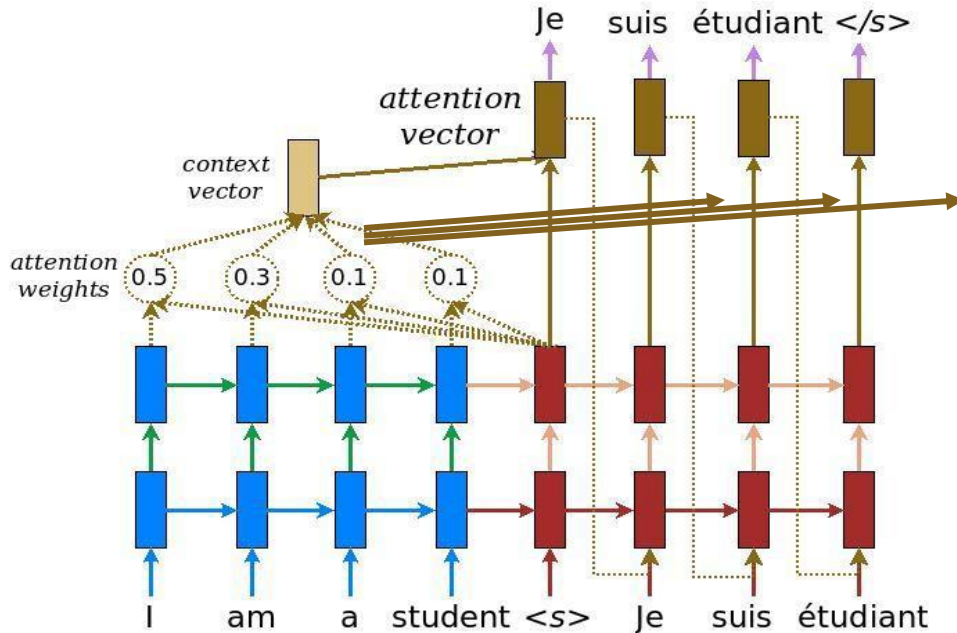
Encoder-Decoder (seq2seq) model

- 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

Encoder-Decoder (seq2seq) model

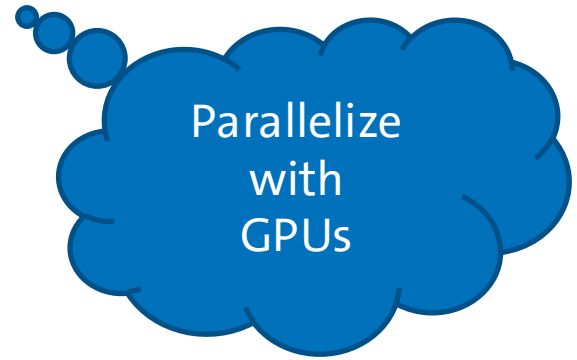


A More General View of seq2seq



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**



Sequence-to-Sequence Approaches

- 2014 – Introduction of Seq2Seq with RNs
 - Model: Encoder–decoder with LSTM
 - Contribution: Introduced the fundamental Seq2Seq paradigm for neural machine translation.
 - Core Idea: An RN encodes the input sequence into a fixed-length context vector, which a second RN decodes into a target sequence.
- Reference:
 - Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. NeurIPS 2014
 - <https://arxiv.org/abs/1409.3215>

Sequence-to-Sequence Approaches

- 2015 – Attention Mechanism
 - Model: Attention-based encoder–decoder
 - Contribution: Enabled the decoder to dynamically focus on different parts of the input sequence, moving beyond the limitations of a fixed context vector.
- Reference:
 - Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate.
 - <https://arxiv.org/abs/1409.0473>

Sequence-to-Sequence Approaches

- 2017 – Transformer: Seq2Seq Without RNs
 - Model: Transformer
 - Contribution: Introduced a fully self-attention-based model, replacing RNNs entirely. Marked a major leap in translation quality, training speed, and context modeling.
- Reference:
 - Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need (2023)
 - <https://arxiv.org/abs/1706.03762>

Sequence-to-Sequence Approaches

- 2020 – BART (Bidirectional and Auto-Regressive Transformers)
 - Model: BART
 - Contribution: Combines an encoder with a decoder. Effective for tasks involving text generation, denoising, and correction.
- Reference:
 - Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension (2019), <https://arxiv.org/abs/1910.13461>

Sequence-to-Sequence Approaches

- 2018 – Marian: Fast Neural Machine Translation in C++
 - Model: MarianMT
 - Contribution: Open-source neural machine translation toolkit designed for speed, efficiency, and multilingual capabilities. Well-suited for low-resource scenarios.
- Reference:
 - Junczys-Dowmunt, M., Grundkiewicz, R., Dwojak, T., Hoang, H., Heafield, K., Neckermann, T., Seide, F., Germann, U., Aji, A.F., Bogoychev, N., Martins, A.F.T., Birch, A.: Marian: Fast neural machine translation in c++ (2018), <https://arxiv.org/abs/1804.00344>

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**

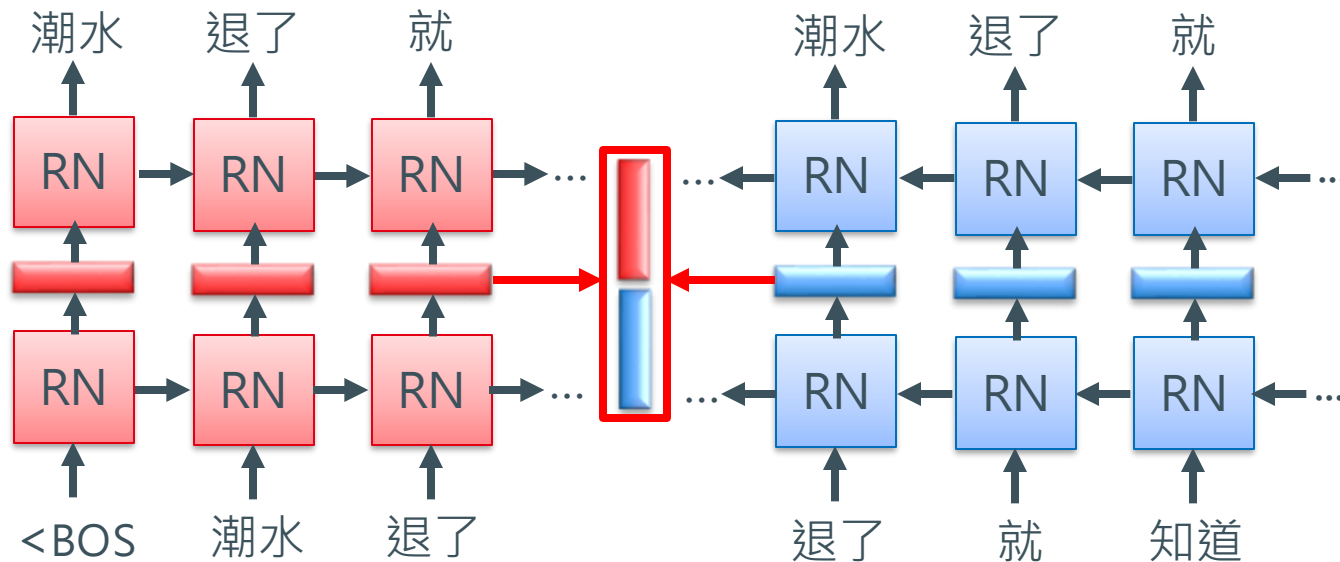
=> Return for each token a **(task-dependent) linear combination of its representation across layers.**

=> Different layers capture different information

Embeddings from **L**anguage **M**odel (ELMO)

- RN-based language models (trained from lots of sentences)

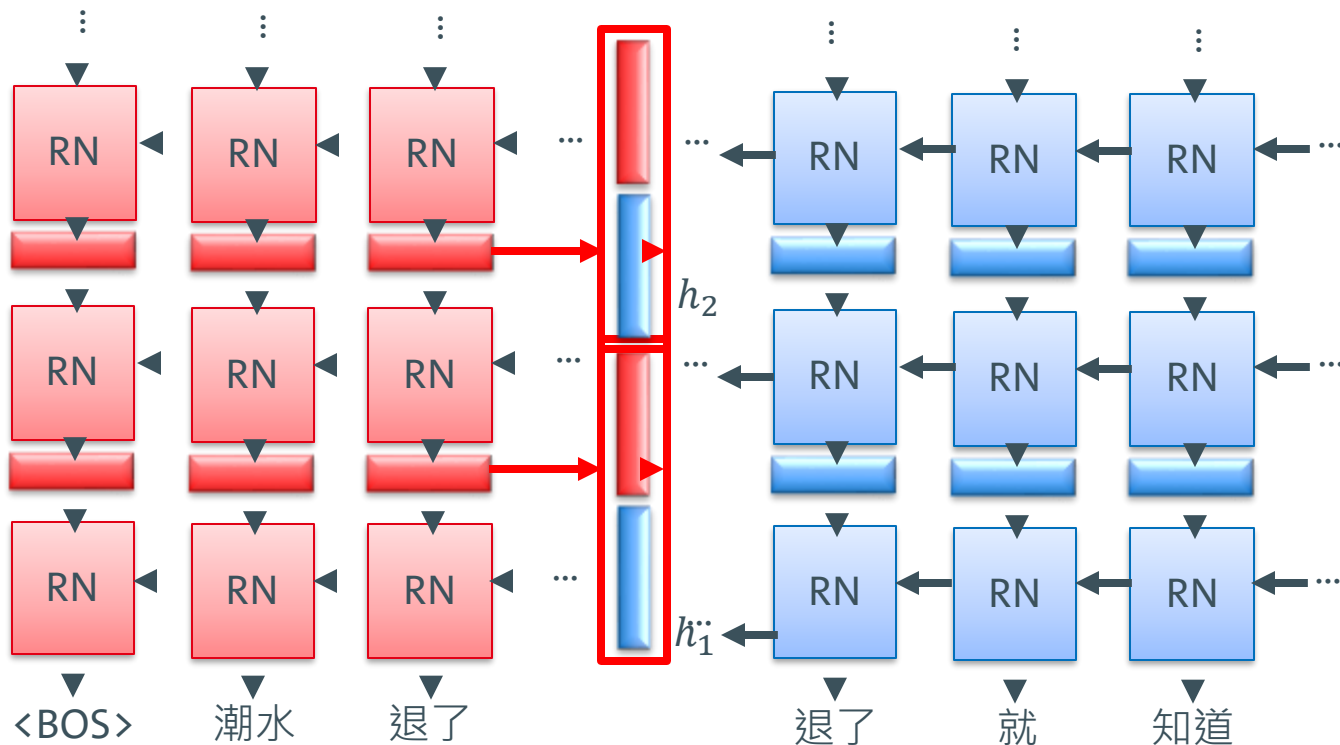
e.g., given “潮水退了就知道誰沒穿褲子”



Each layer in deep LSTM can generate a latent representation.

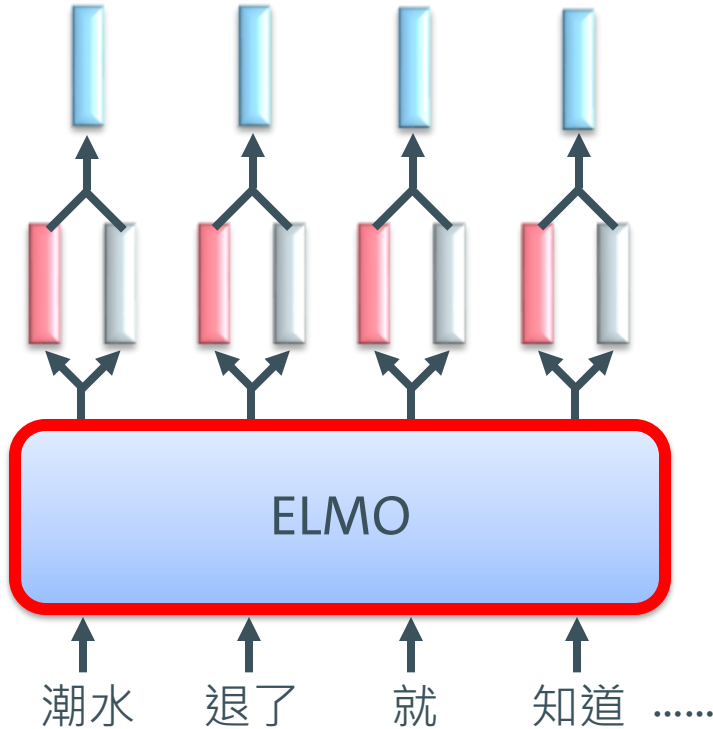
Which one should we use???

ELMO



ELMO

High computational effort, word2vec to the rescue?

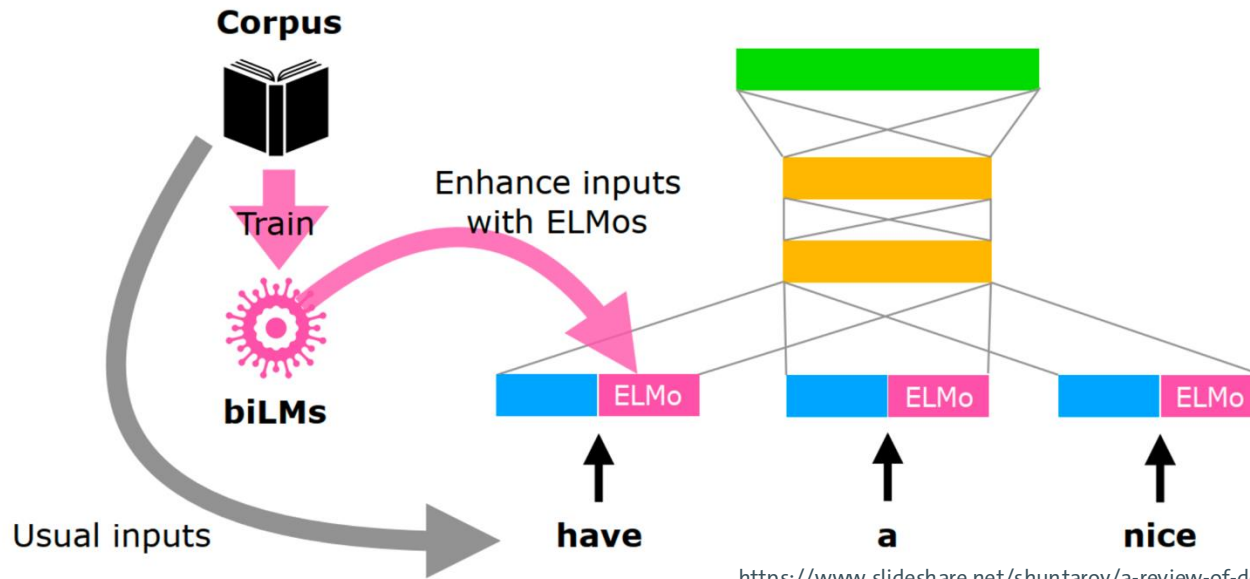


$$\text{Blue Embedding} = \alpha_1 \text{Red Embedding} + \alpha_2 \text{Grey Embedding}$$

Learned with the down-stream tasks

http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML20.html

Integrate ELMos into other embeddings



<https://www.slideshare.net/shuntaroy/a-review-of-deep-contextualized-word-representations-peters-2018>

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 sub-words 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**.

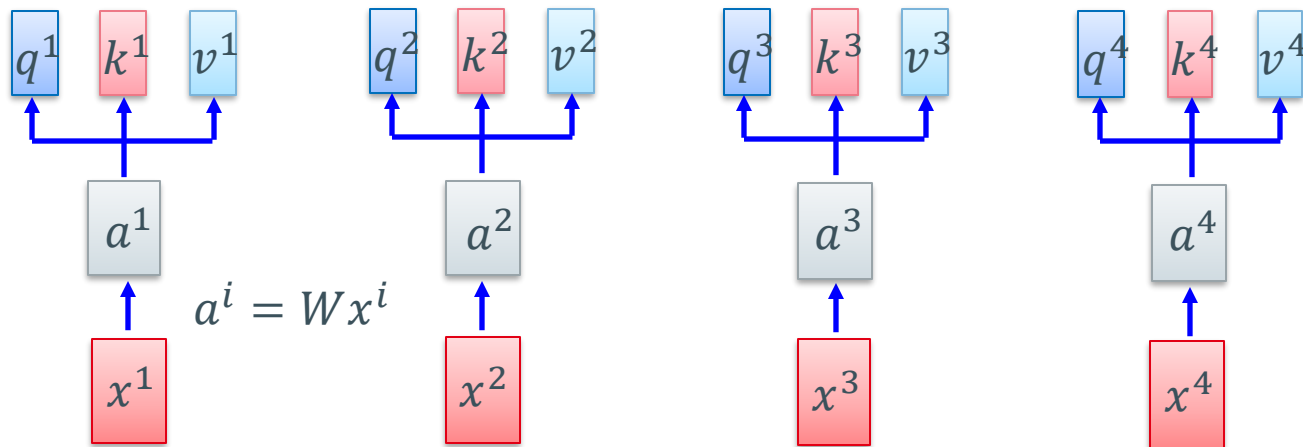
Self-attention

q : query (to match others) k : key (to be matched) v : information to be extracted

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

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

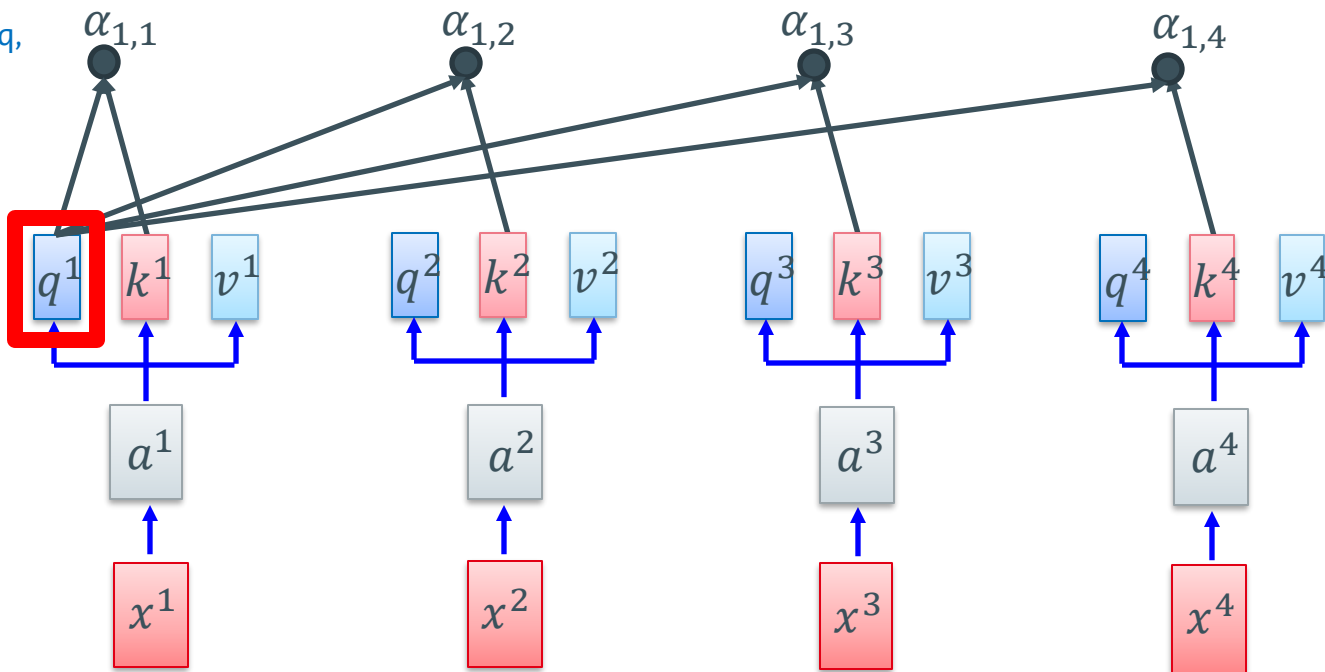
$$v^i = W^v a^i$$



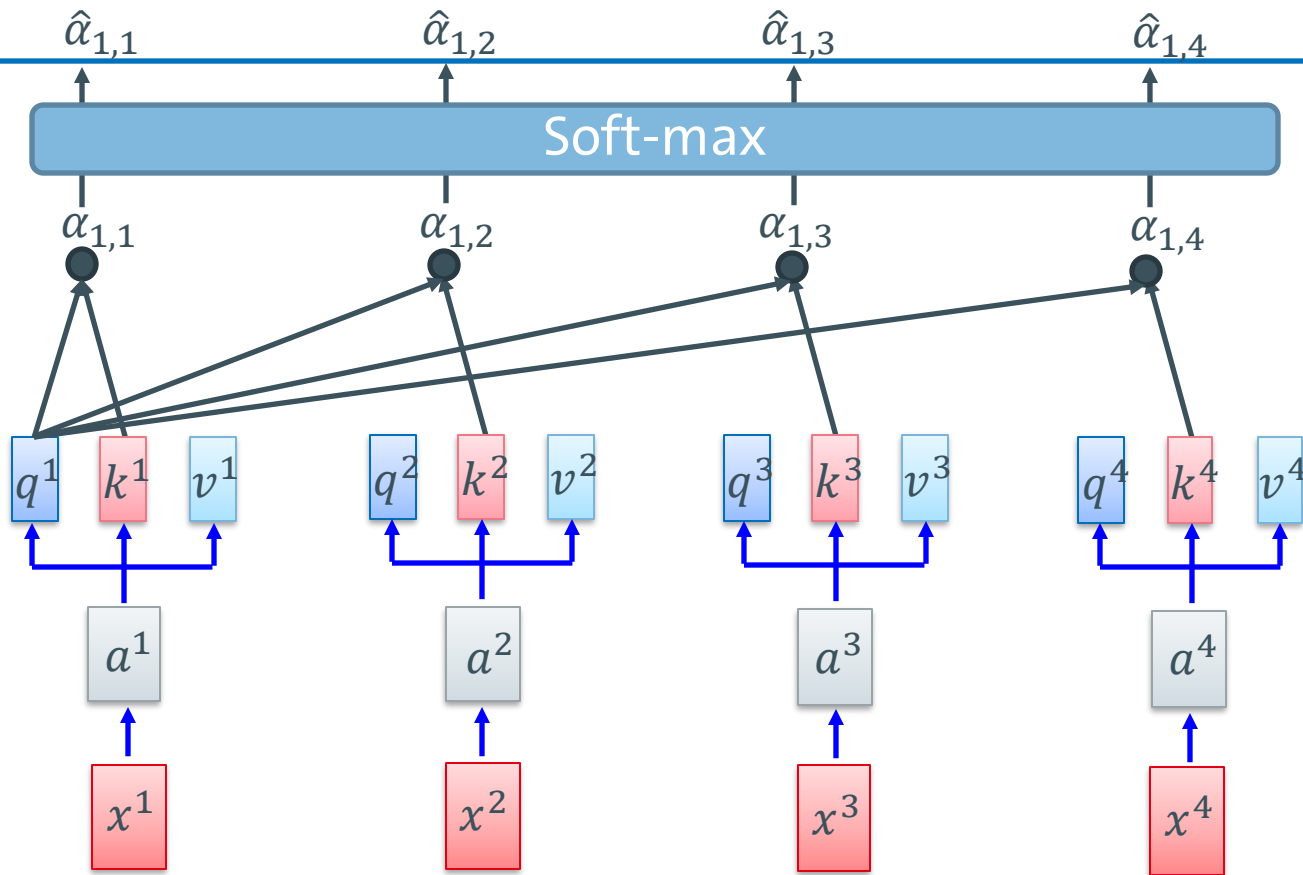
Scaled Dot-Product Attention: $\alpha_{1,i} = \underbrace{q^1 \cdot k^i}_{\text{dot product}} / \sqrt{d}$

Self-attention

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

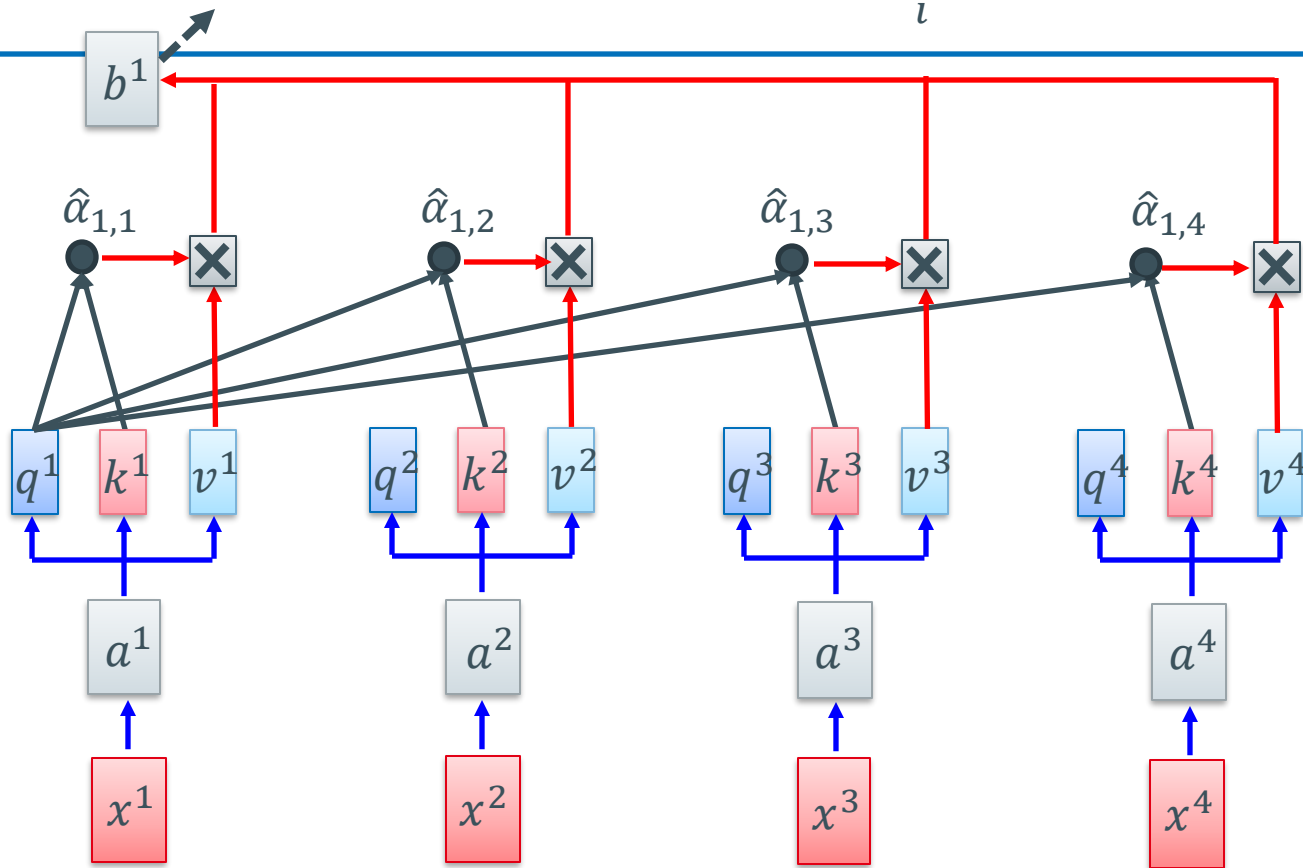


$$\hat{\alpha}_{1,i} = \exp(\alpha_{1,i}) / \sum_j \exp(\alpha_{1,j})$$



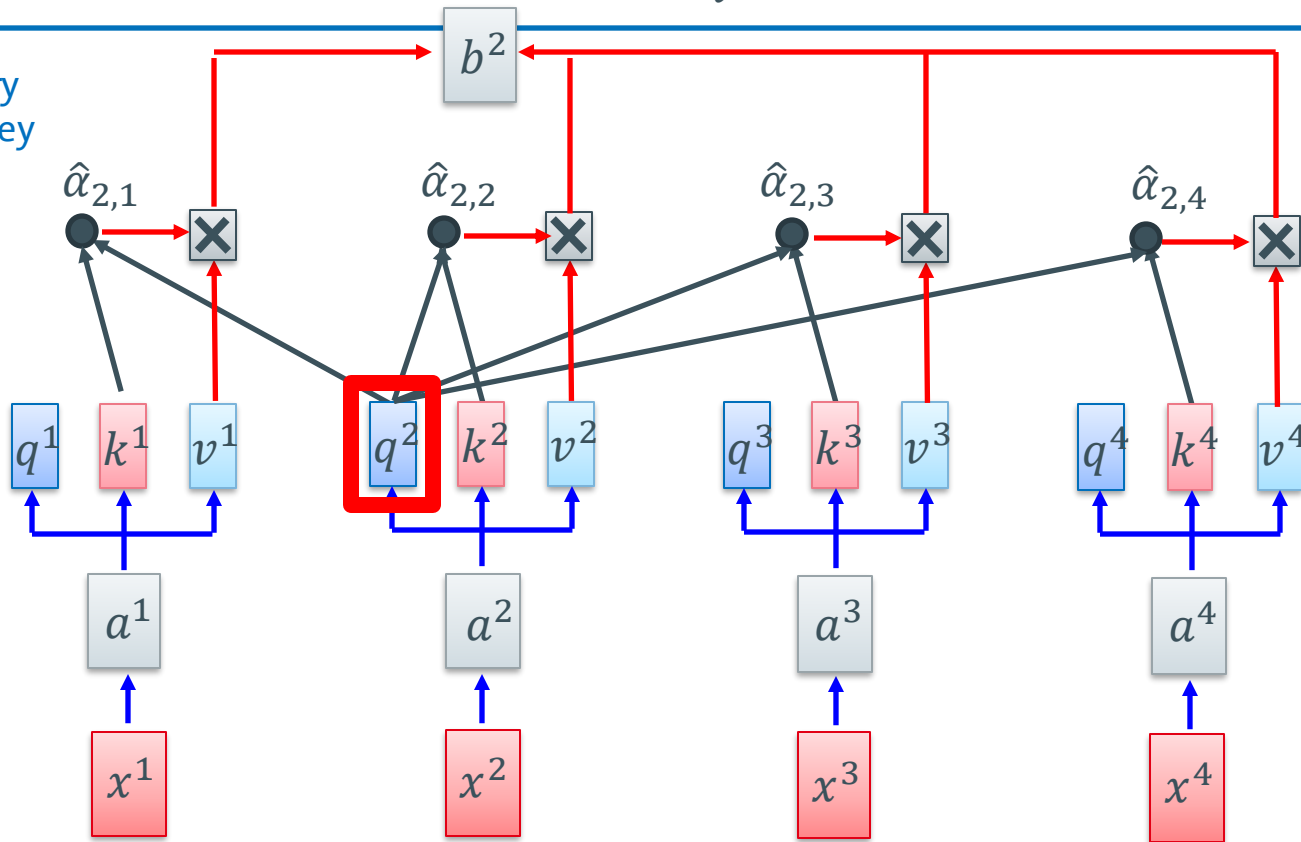
Considering the whole sequence

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

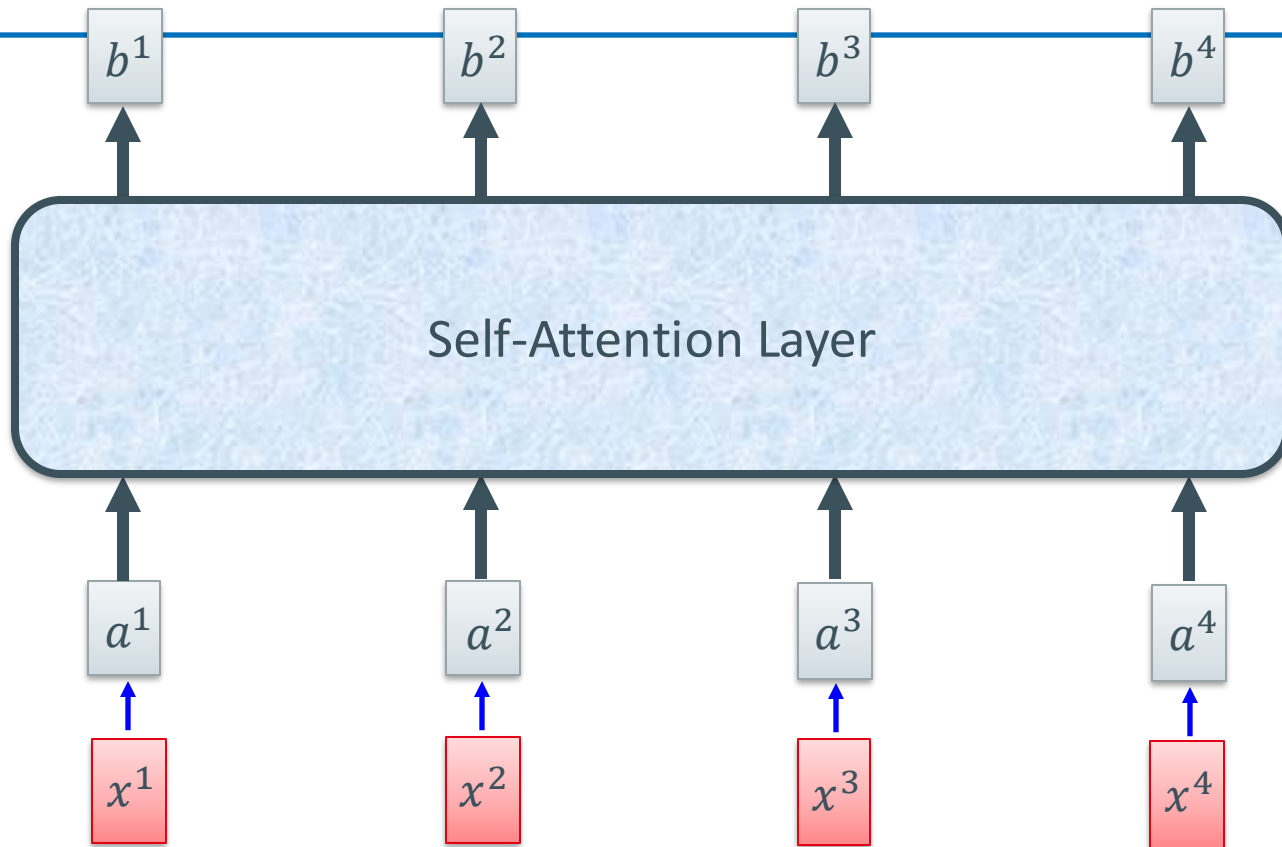


$$b^2 = \sum_i \hat{\alpha}_{2,i} v^i$$

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



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



$$\begin{array}{c}
 \boxed{q^1} \boxed{q^2} \boxed{q^3} \boxed{q^4} \\
 Q
 \end{array}
 = \boxed{W^q} \begin{array}{c}
 \boxed{a^1} \boxed{a^2} \boxed{a^3} \boxed{a^4} \\
 I
 \end{array}$$

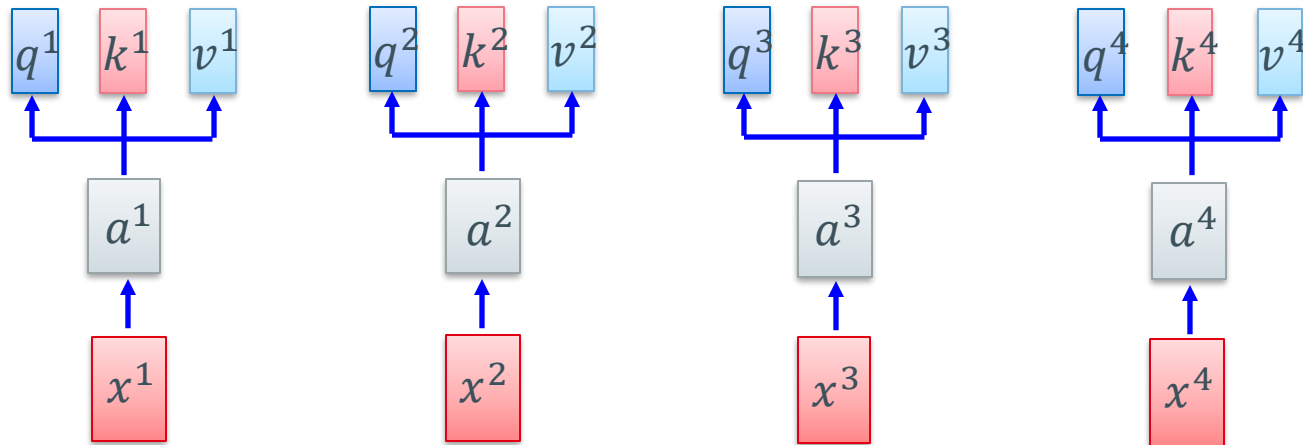
$$q^i = W^q a^i$$

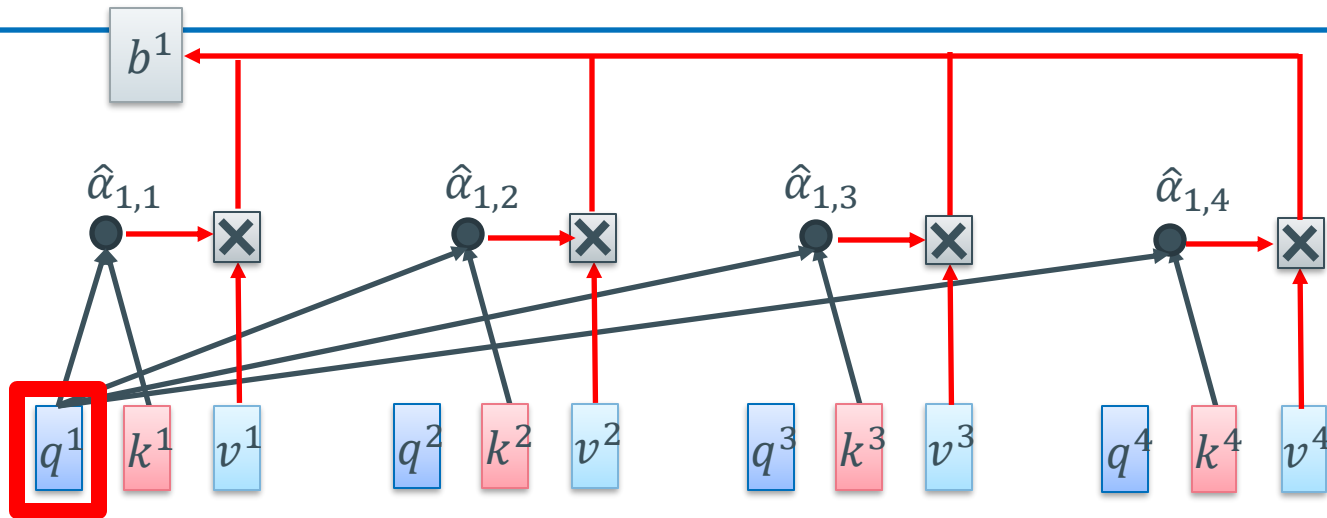
$$k^i = W^k a^i$$

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

$$\begin{array}{c}
 \boxed{k^1} \boxed{k^2} \boxed{k^3} \boxed{k^4} \\
 K
 \end{array}
 = \boxed{W^k} \begin{array}{c}
 \boxed{a^1} \boxed{a^2} \boxed{a^3} \boxed{a^4} \\
 I
 \end{array}$$

$$\begin{array}{c}
 \boxed{v^1} \boxed{v^2} \boxed{v^3} \boxed{v^4} \\
 V
 \end{array}
 = \boxed{W^v} \begin{array}{c}
 \boxed{a^1} \boxed{a^2} \boxed{a^3} \boxed{a^4} \\
 I
 \end{array}$$



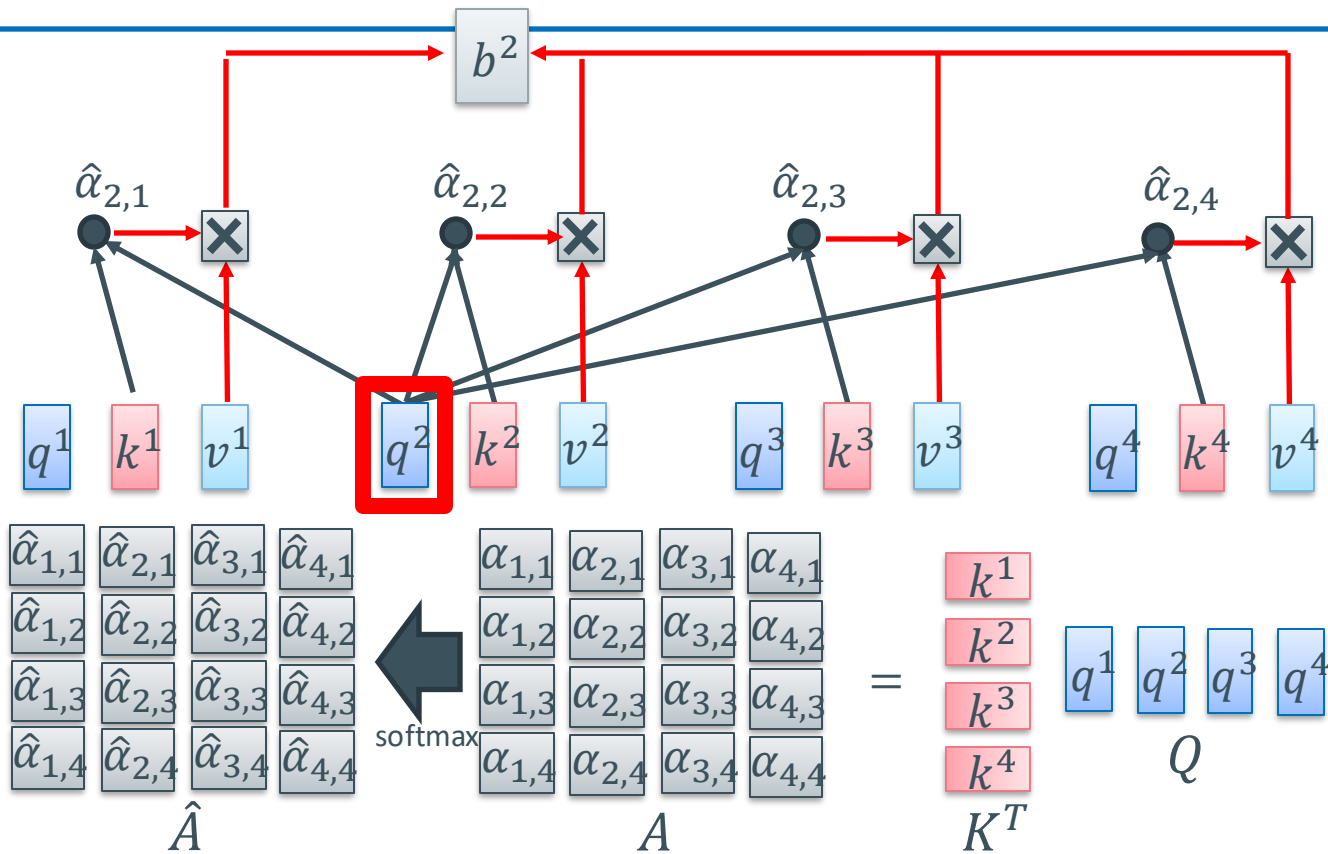


$$\begin{aligned}
 \alpha_{1,1} &= k^1 q^1 & \alpha_{1,2} &= k^2 q^1 \\
 \alpha_{1,3} &= k^3 q^1 & \alpha_{1,4} &= k^4 q^1
 \end{aligned}$$

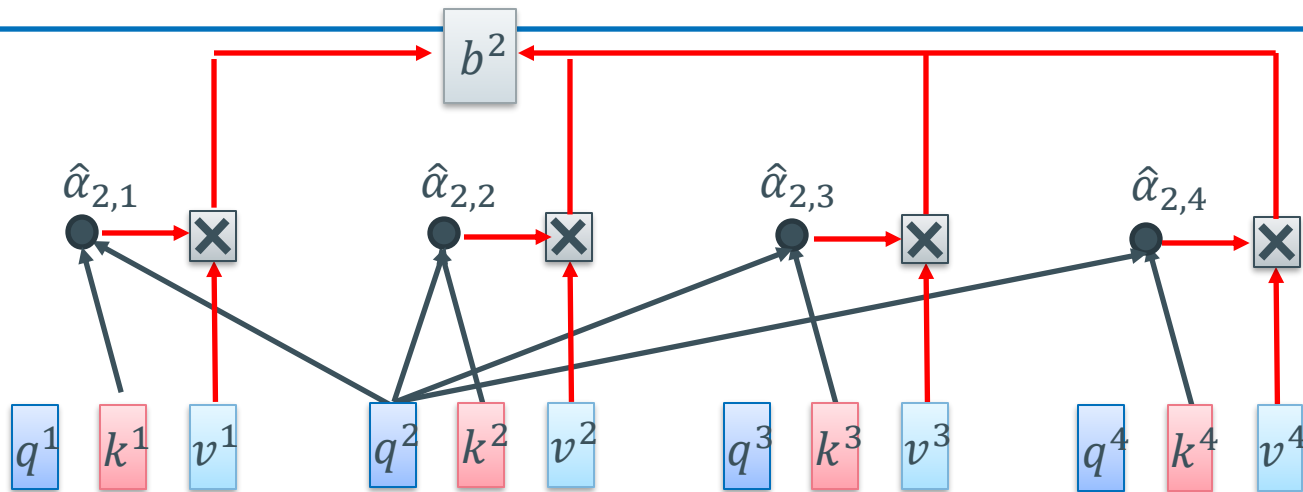
$$\begin{bmatrix} \alpha_{1,1} \\ \alpha_{1,2} \\ \alpha_{1,3} \\ \alpha_{1,4} \end{bmatrix} = \begin{bmatrix} k^1 \\ k^2 \\ k^3 \\ k^4 \end{bmatrix} q^1$$

WiSe2025-2026 (ignore \sqrt{d} for simplicity)

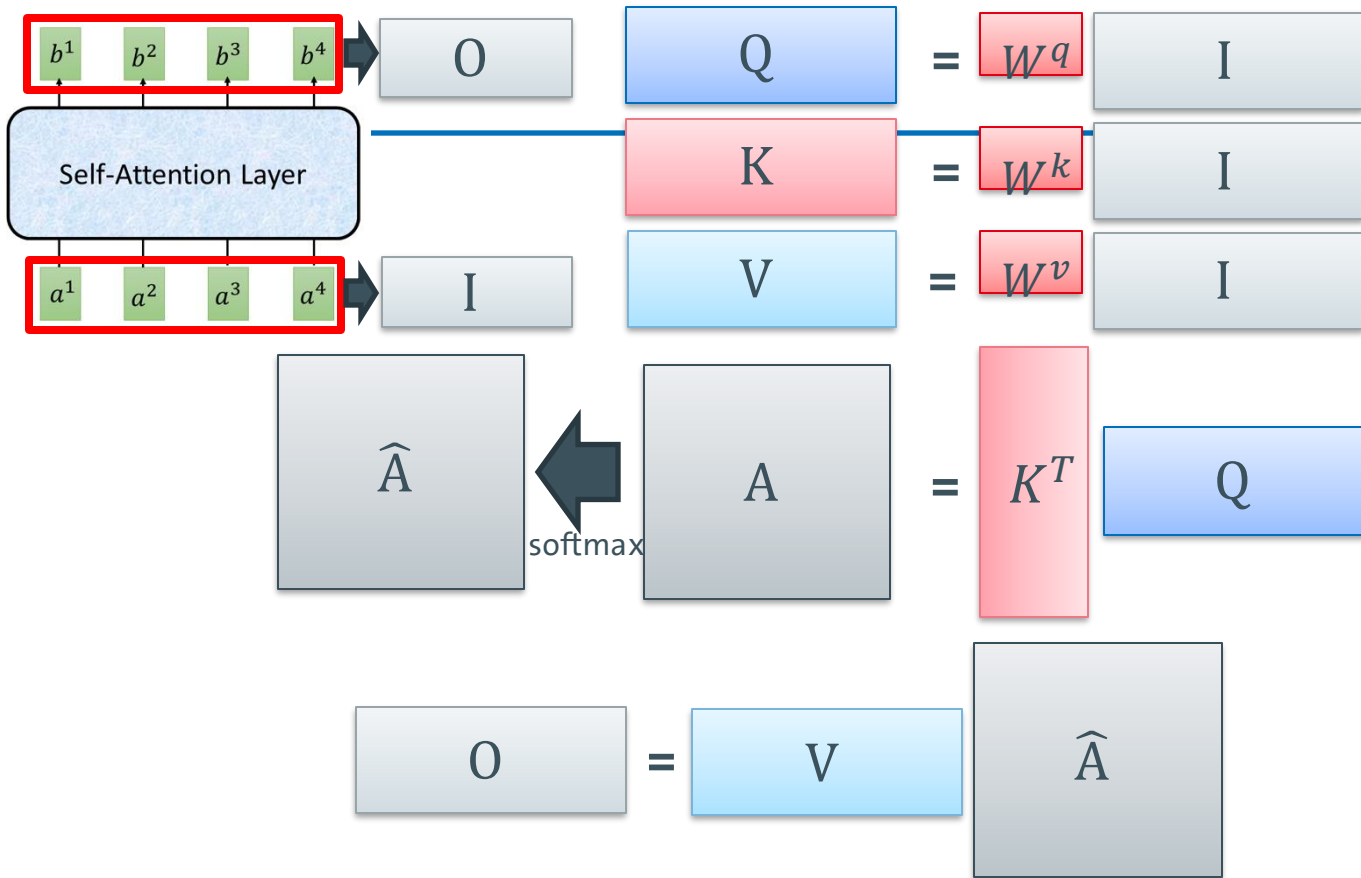
$$b^2 = \sum_i \hat{\alpha}_{2,i} v^i$$



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



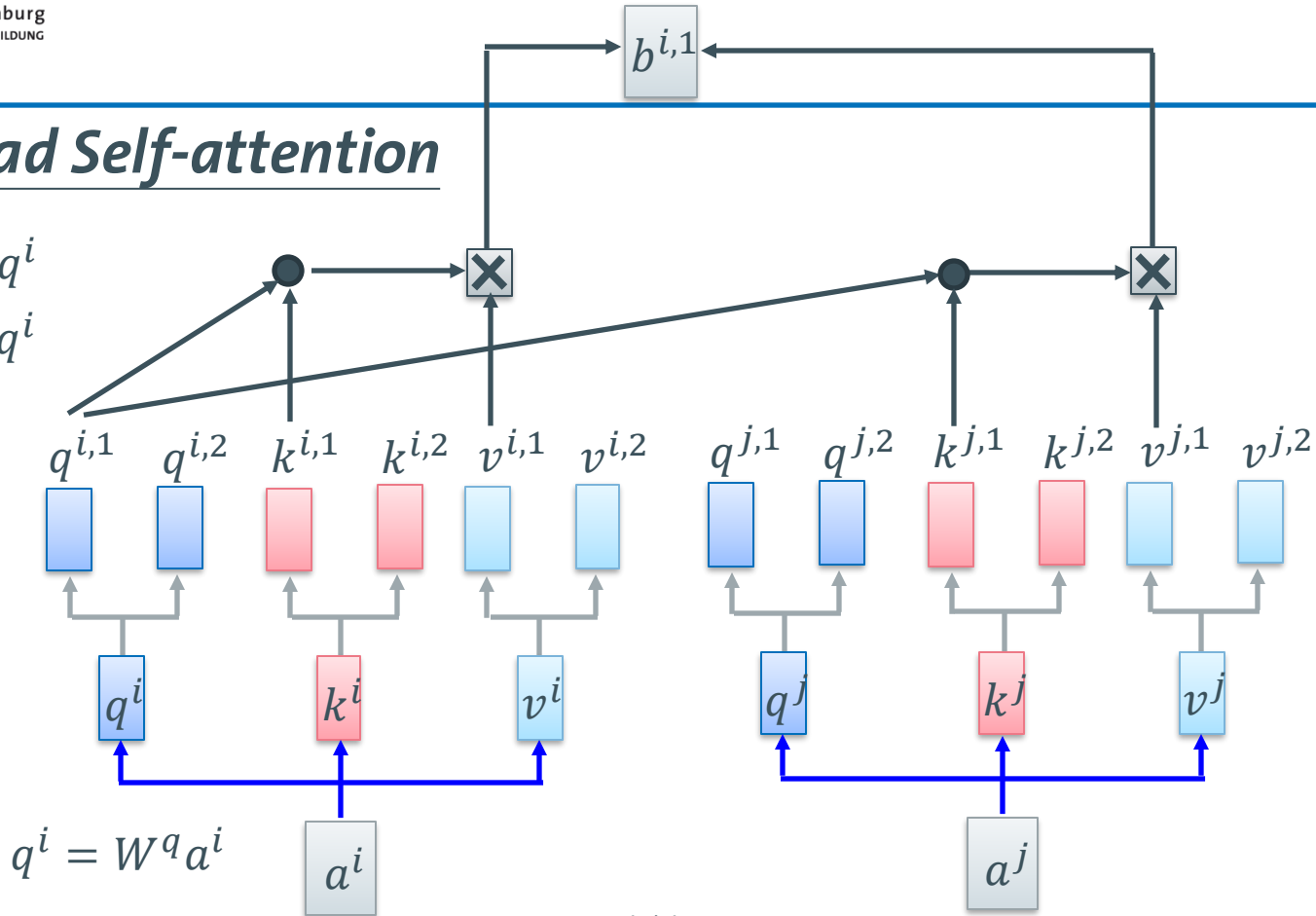
$$\begin{matrix} b^1 & b^2 & b^3 & b^4 \\ \hline \end{matrix}
 \quad = \quad
 \begin{matrix} v^1 & v^2 & v^3 & v^4 \\ \hline \end{matrix}
 \quad
 \begin{matrix} \hat{\alpha}_{1,1} & \hat{\alpha}_{2,1} & \hat{\alpha}_{3,1} & \hat{\alpha}_{4,1} \\ \hat{\alpha}_{1,2} & \hat{\alpha}_{2,2} & \hat{\alpha}_{3,2}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



Multi-head Self-attention

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

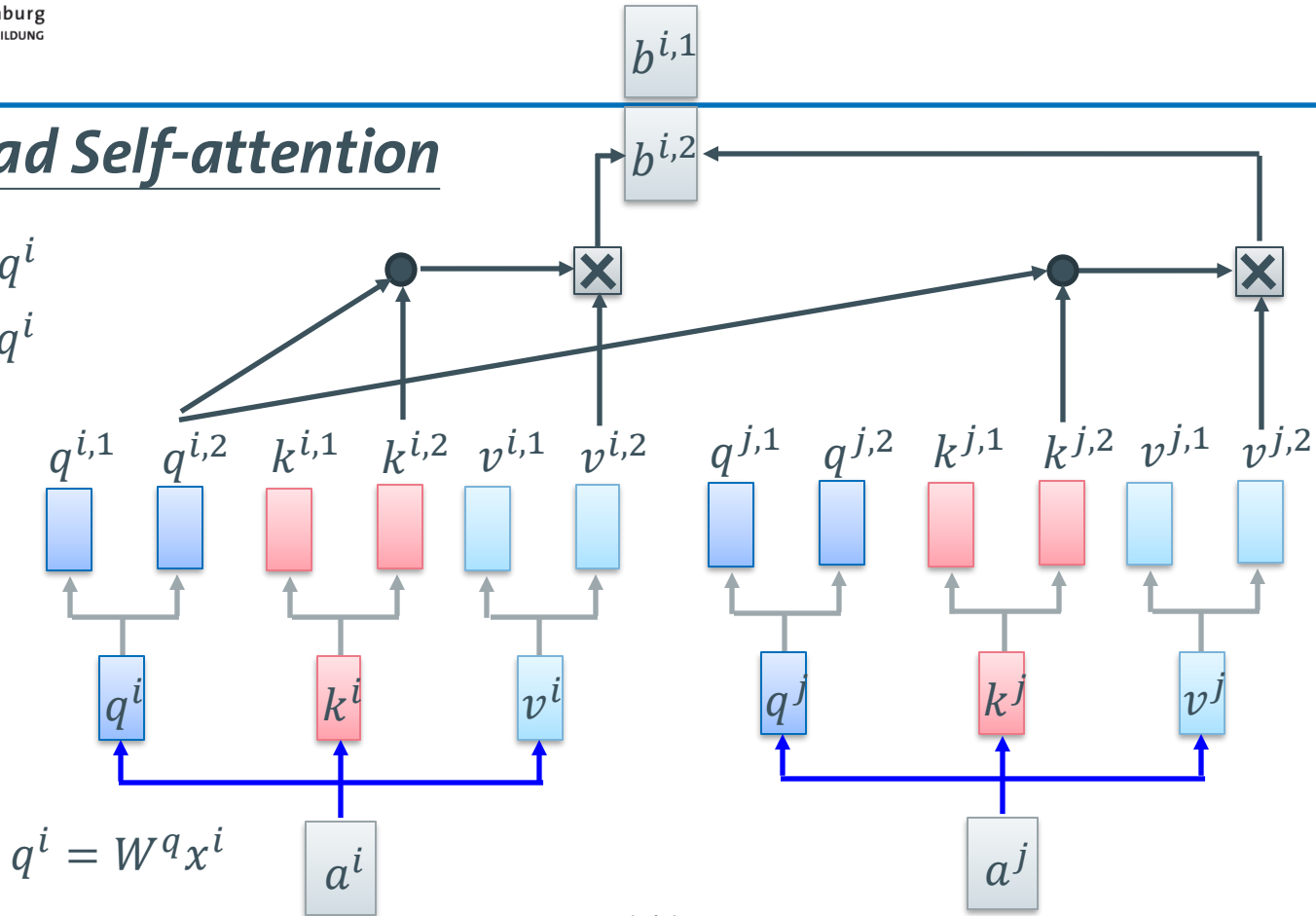
$$q^{i,2} = W^{q,2} q^i$$



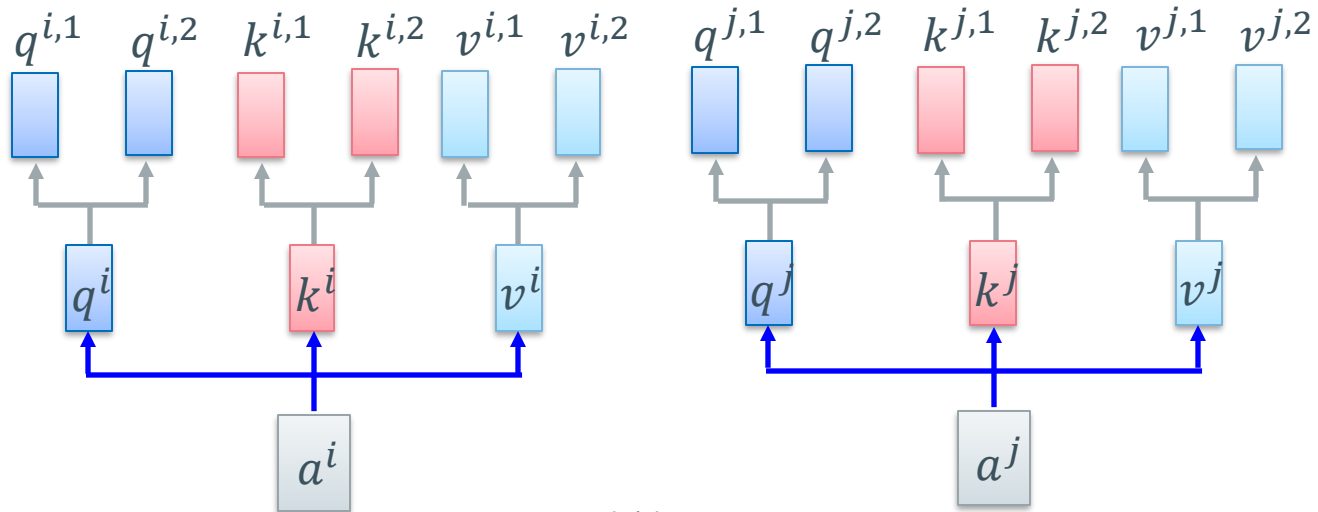
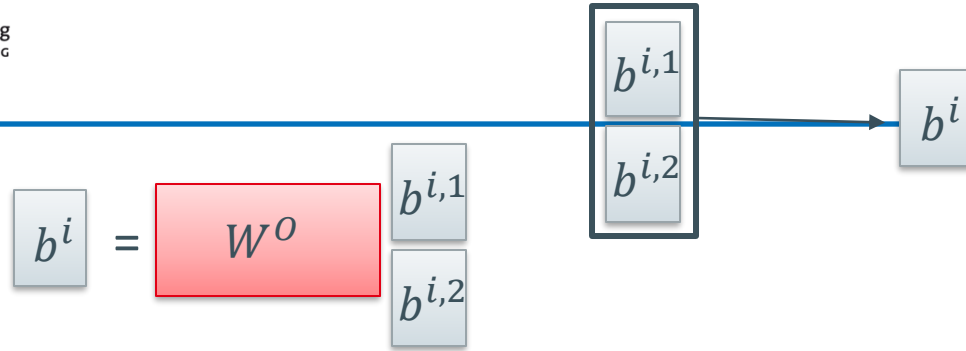
Multi-head Self-attention

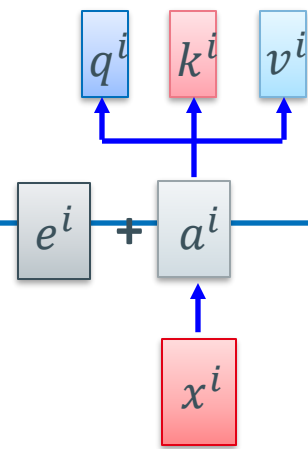
$$q^{i,1} = W^{q,1} q^i$$

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



(2 heads as example)





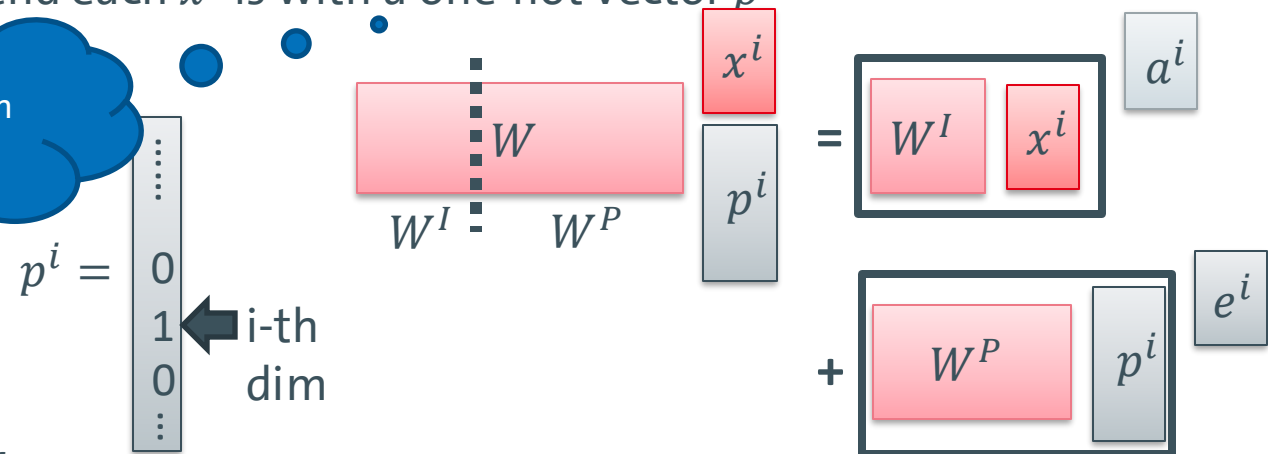
Positional Encoding

No position information in self-attention.

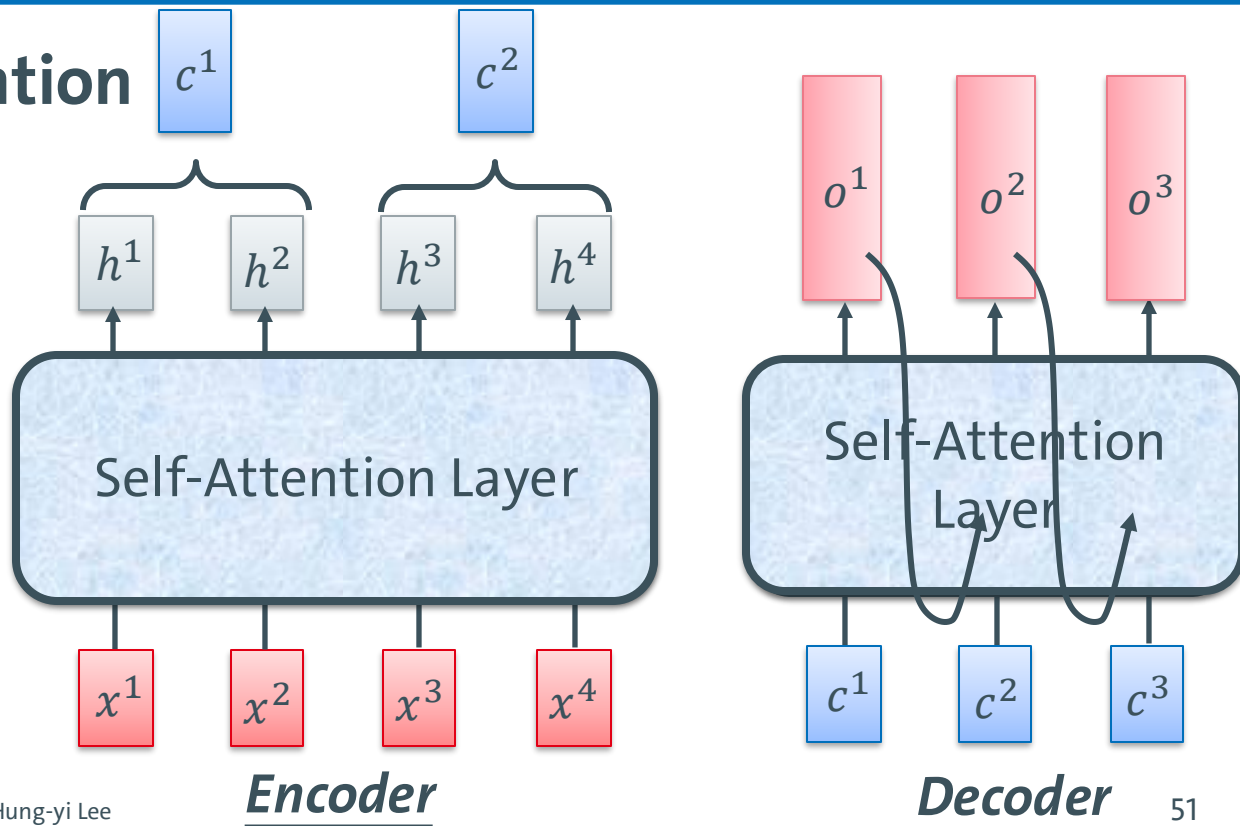
Each position has a unique positional vector e^i (not learned from data)

Idea: Append each x^i is with a one-hot vector p^i

More clever solution used in the original paper



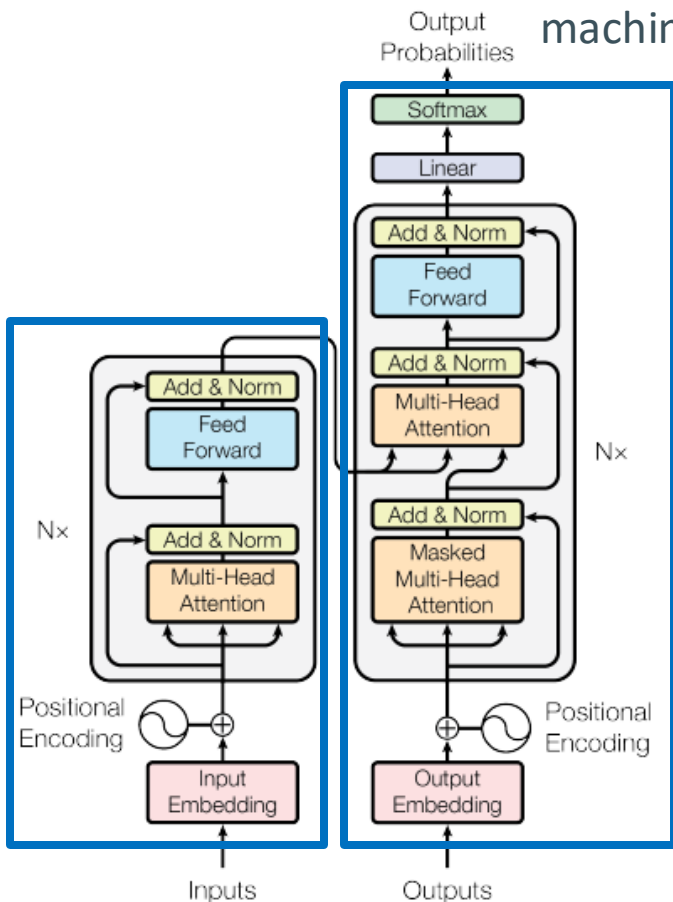
Seq2seq with Attention



Transformer

Using Chinese to English translation as example

Encoder



Decoder

機器學習

<BOS>

machine

Transformer

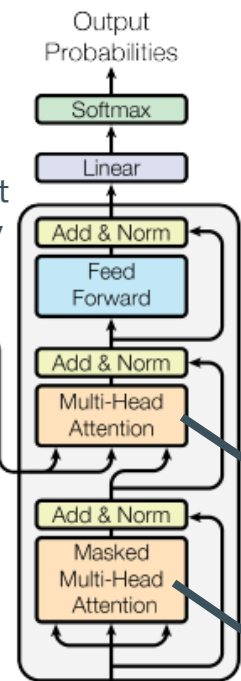
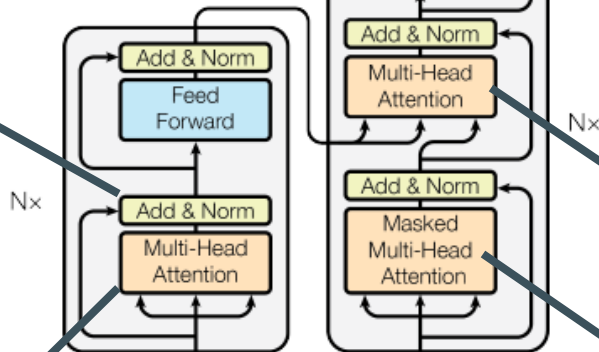
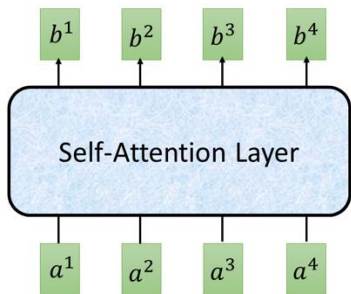
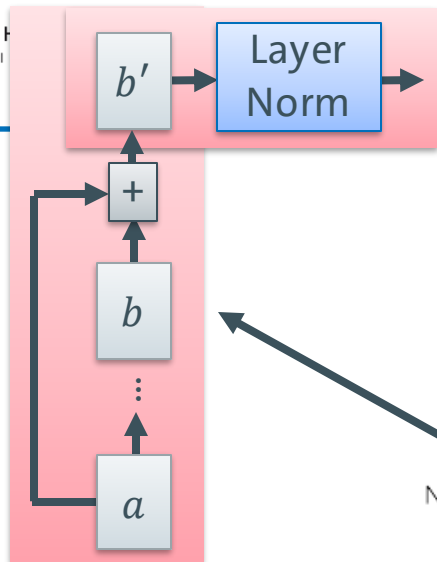
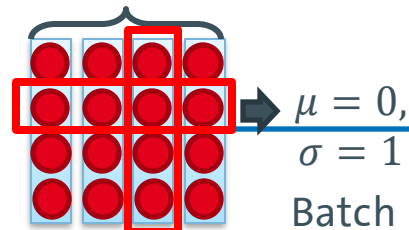
Layer Norm:

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

Batch Norm:

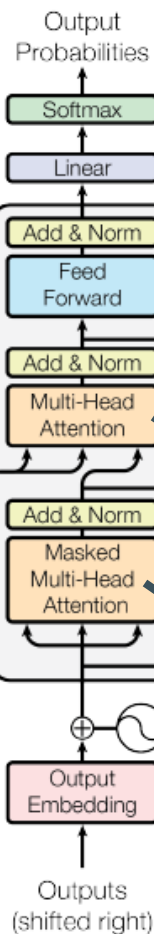
<https://www.youtube.com/watch?v=BZhl1tr5Rkg>

Batch Size



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

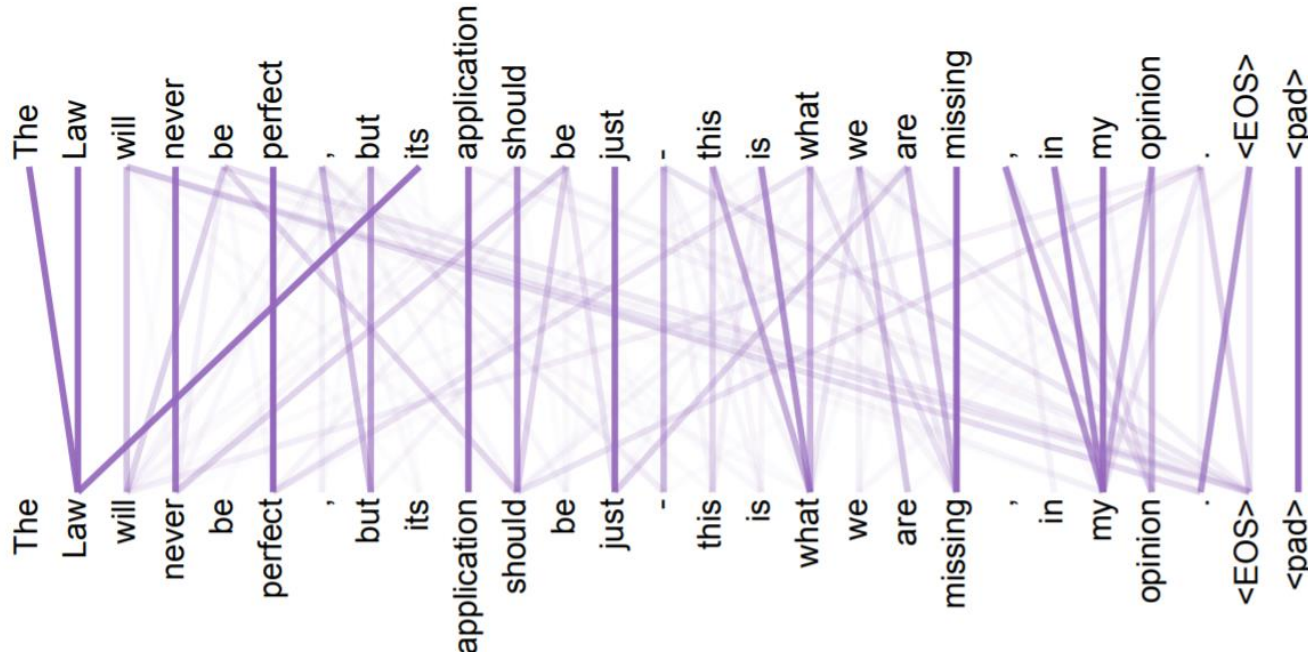
Copy output #token times

For each position use [MASK] token in copies

Attention becomes possible during training also for decoding

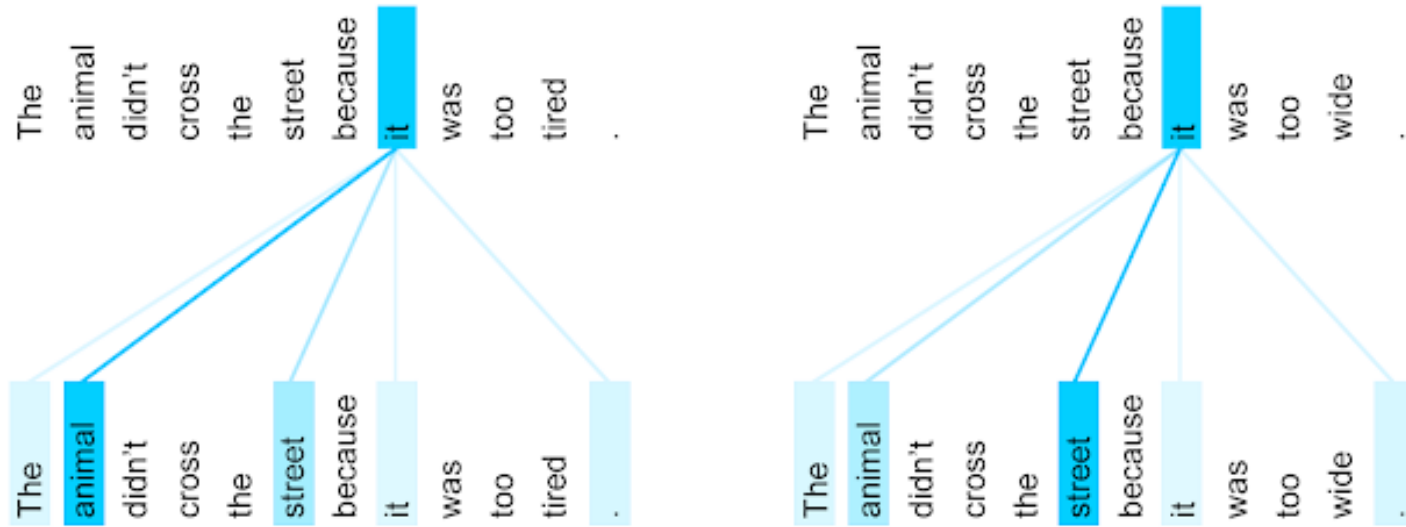
Train decoder such that [MASK] is replaced correctly while paying attention to the overall output training data

Attention Visualization



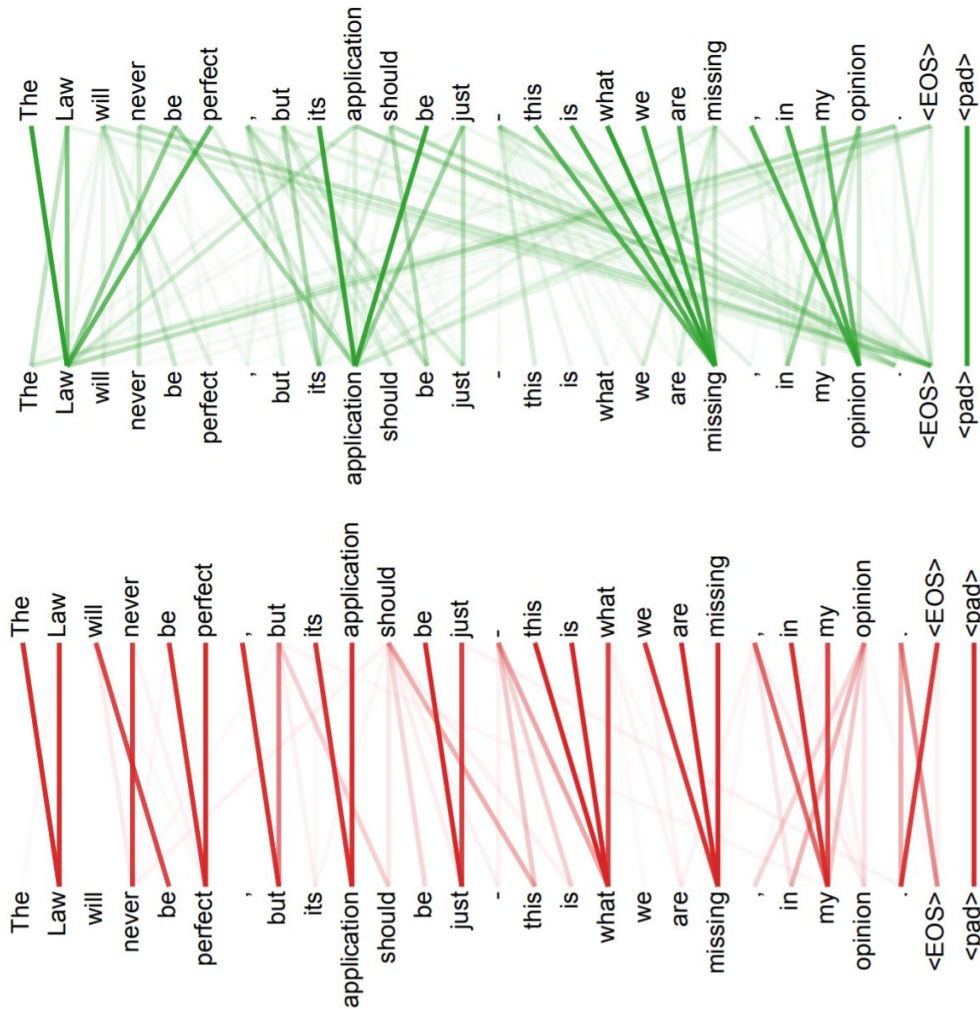
<https://arxiv.org/abs/1706.03762>

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).

Multi-head Attention



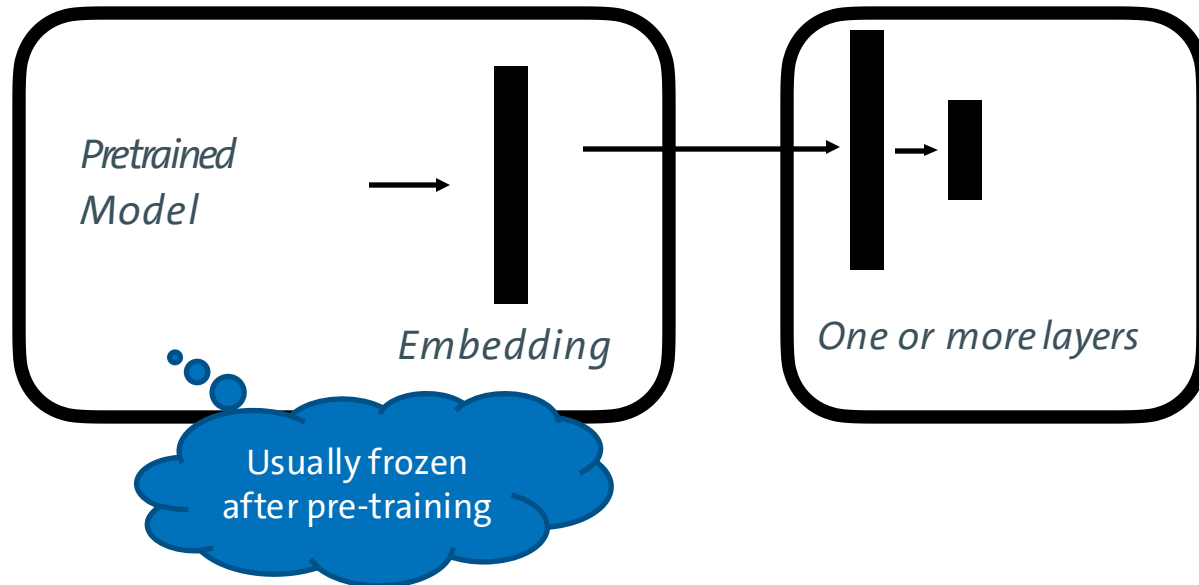
Pre-Training & Fine Tuning

1) Download LM

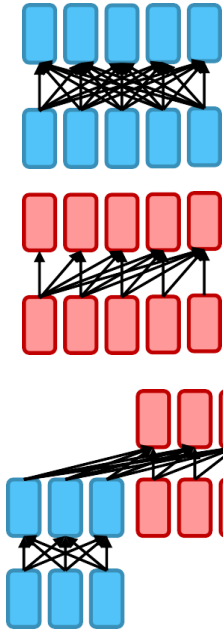
pre-trained on large corpus
(in self-supervised fashion)

2) Feature-based training ("fine-tuning")

on target task
(supervised learning)



Model Pre-Training



Encoder

- Bidirectional context
- Examples: BERT and its variants

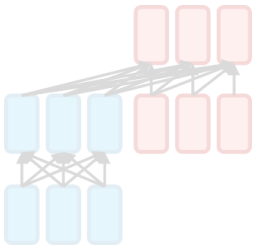
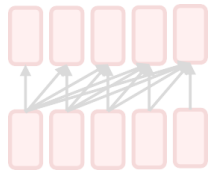
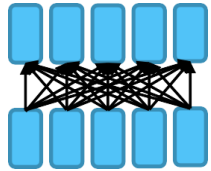
Decoder

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

Encoder-Decoder

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

Model Pre-Training

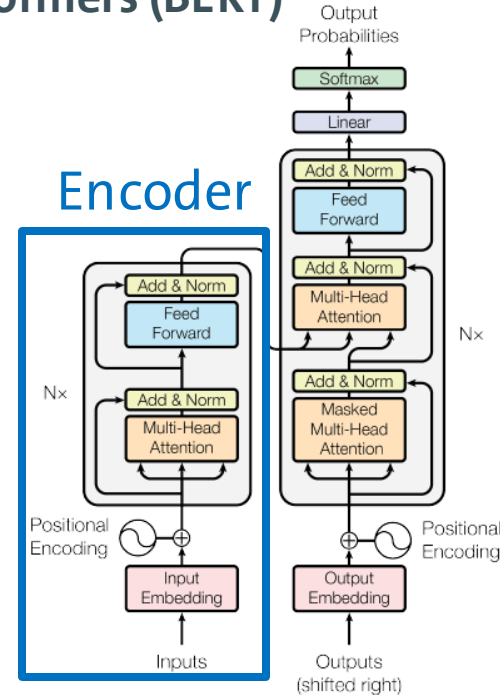
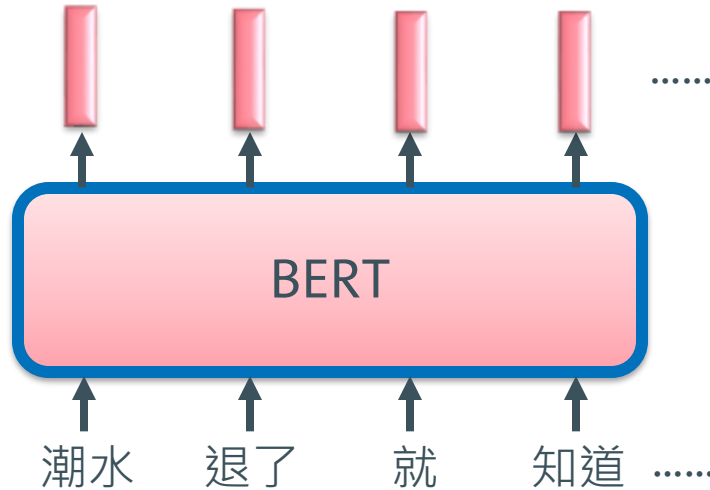


- Encoder
 - Bidirectional context
 - Examples: BERT and its variants
- Decoder
 - Language modeling; better for generation
 - Example: GPT-2, GPT-3, [LaMDA](#)
- Encoder-Decoder
 - Sequence-to-sequence model
 - Examples: Transformer, BART, T5

Bidirectional Encoder Representations from Transformers (BERT)

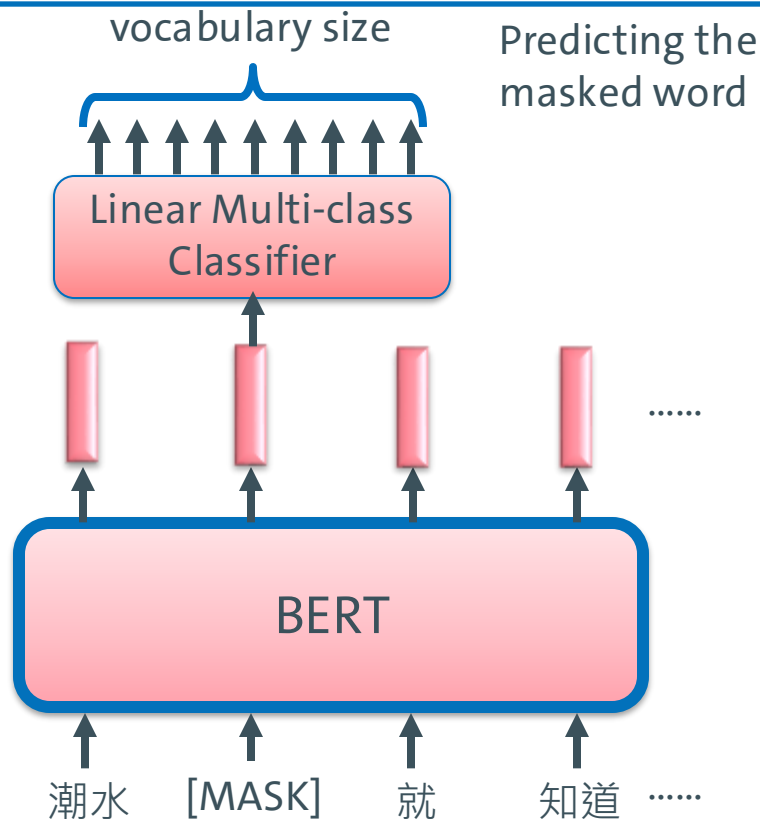
BERT = Encoder of Transformer

Learned from a large amount of text
without annotation

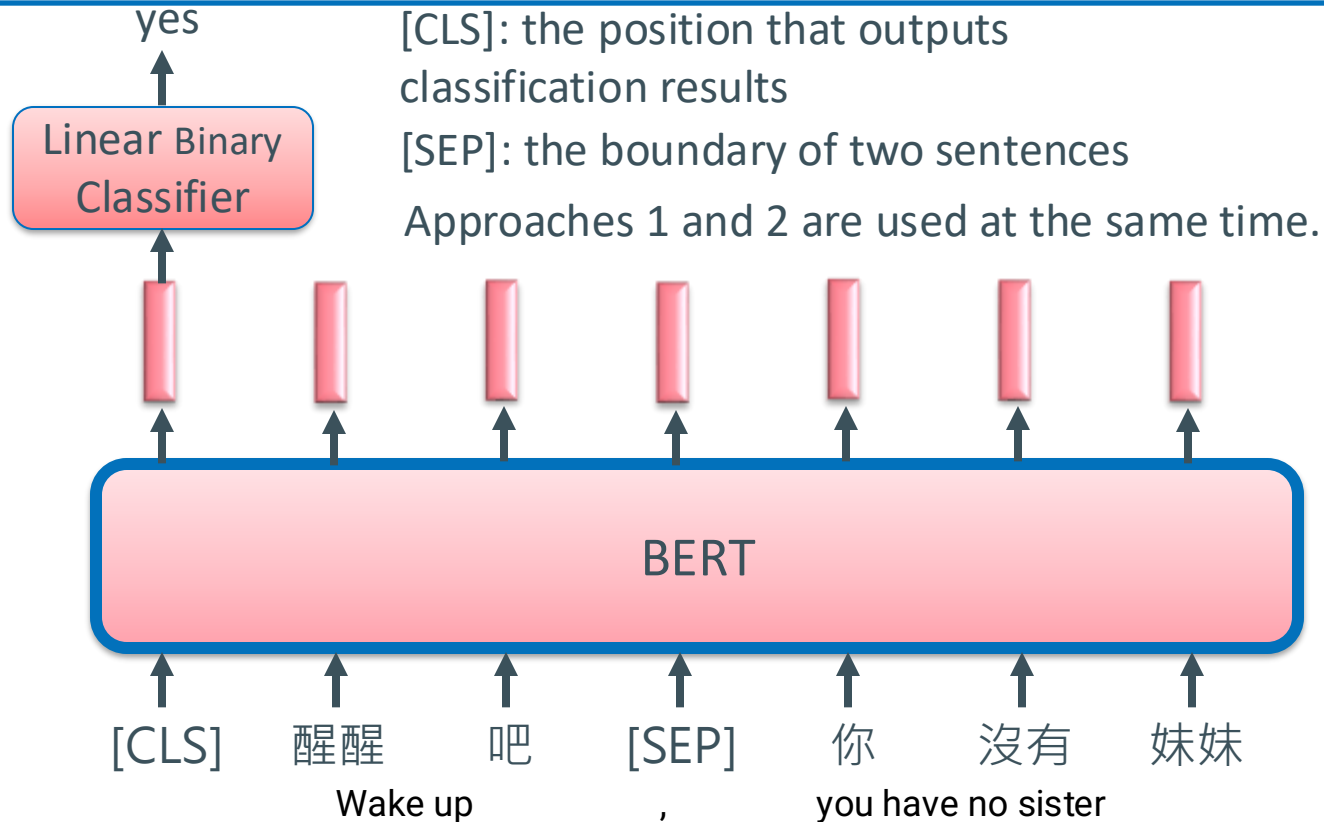


Training of BERT

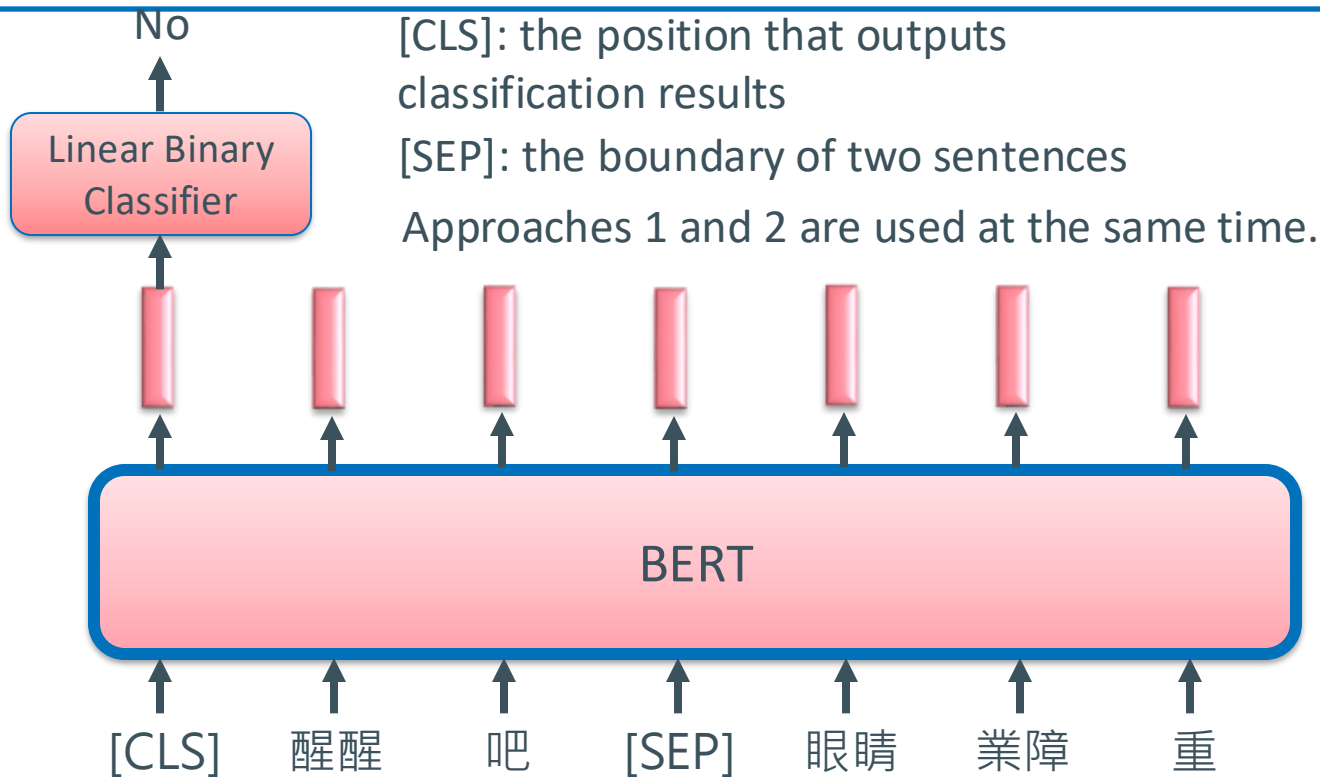
Approach 1: Masked LM



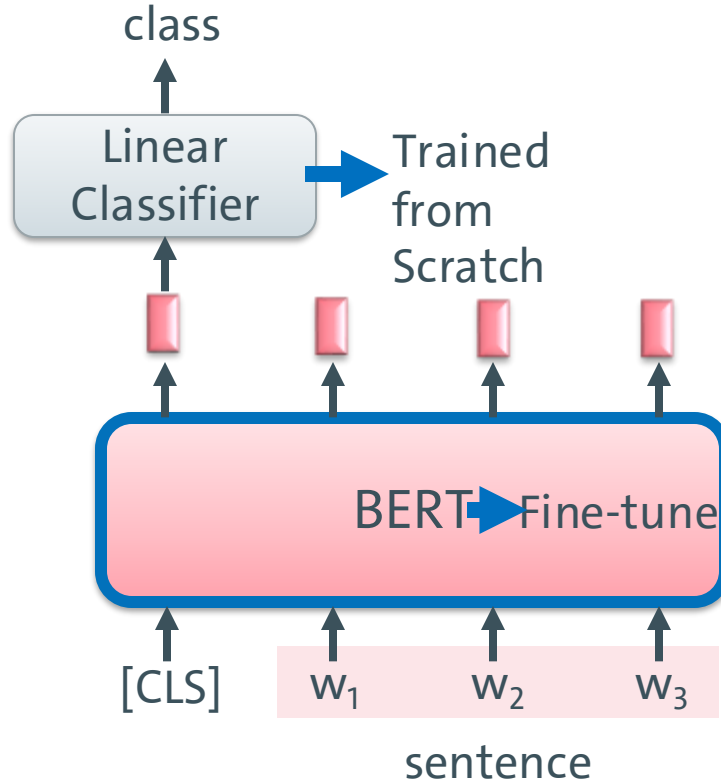
Training of BERT – Approach 2: Next Sentence Prediction



Training of BERT – Approach 2: Next Sentence Prediction



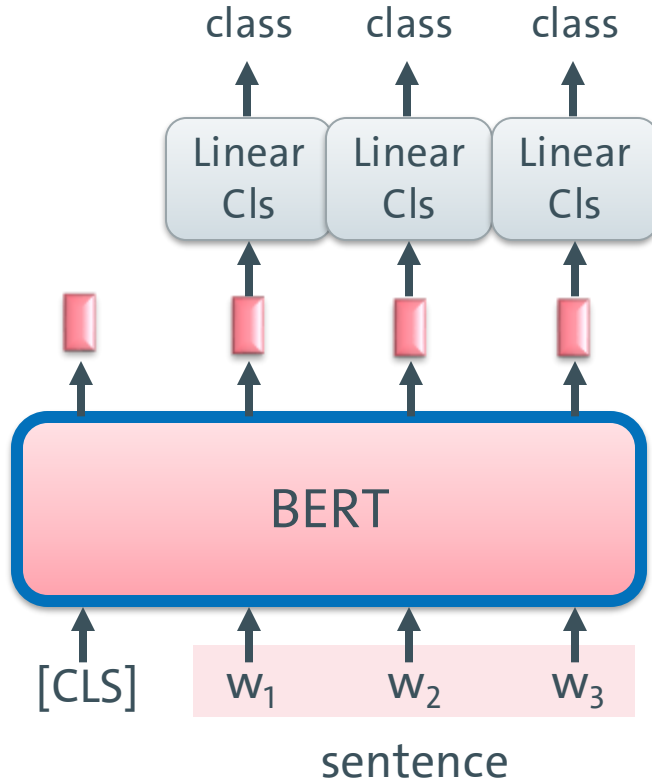
How to use BERT – Case 1



Input: single sentence,
output: class

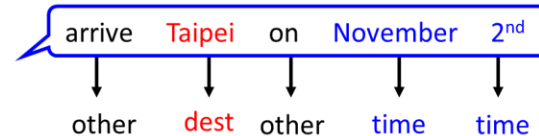
Example:
Sentiment analysis,
Document
Classification

How to use BERT – Case 2

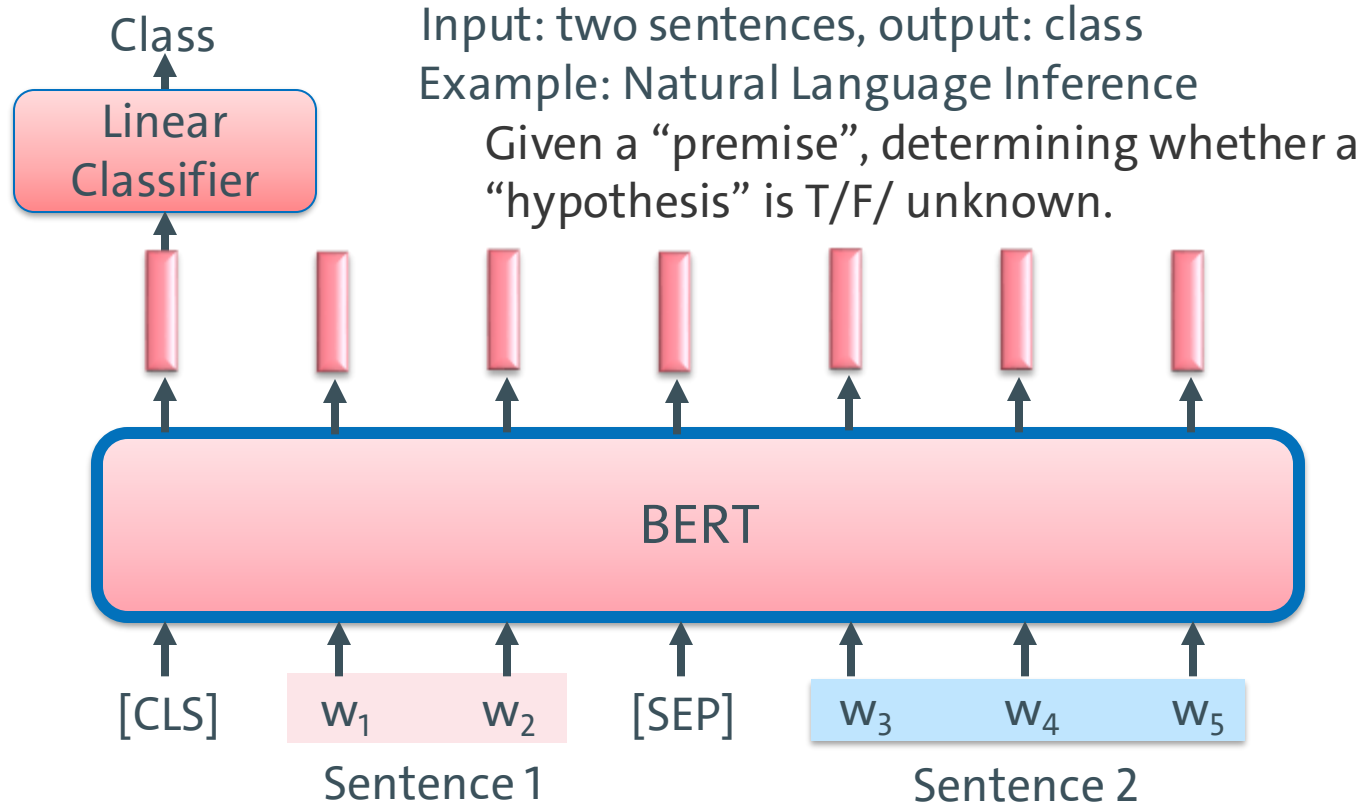


Input: single sentence,
output: class of each word

Example: Semantic
role labelling



How to use BERT – Case 3

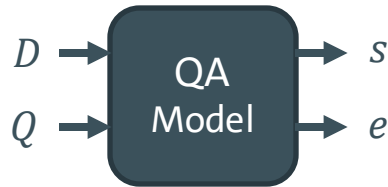


How to use BERT – Case 4

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, **graupel** and hail... Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals **within a cloud**. Short, intense periods of rain 77 are called "showers". 79

What causes precipitation to fall?

gravity s = 17, e = 17

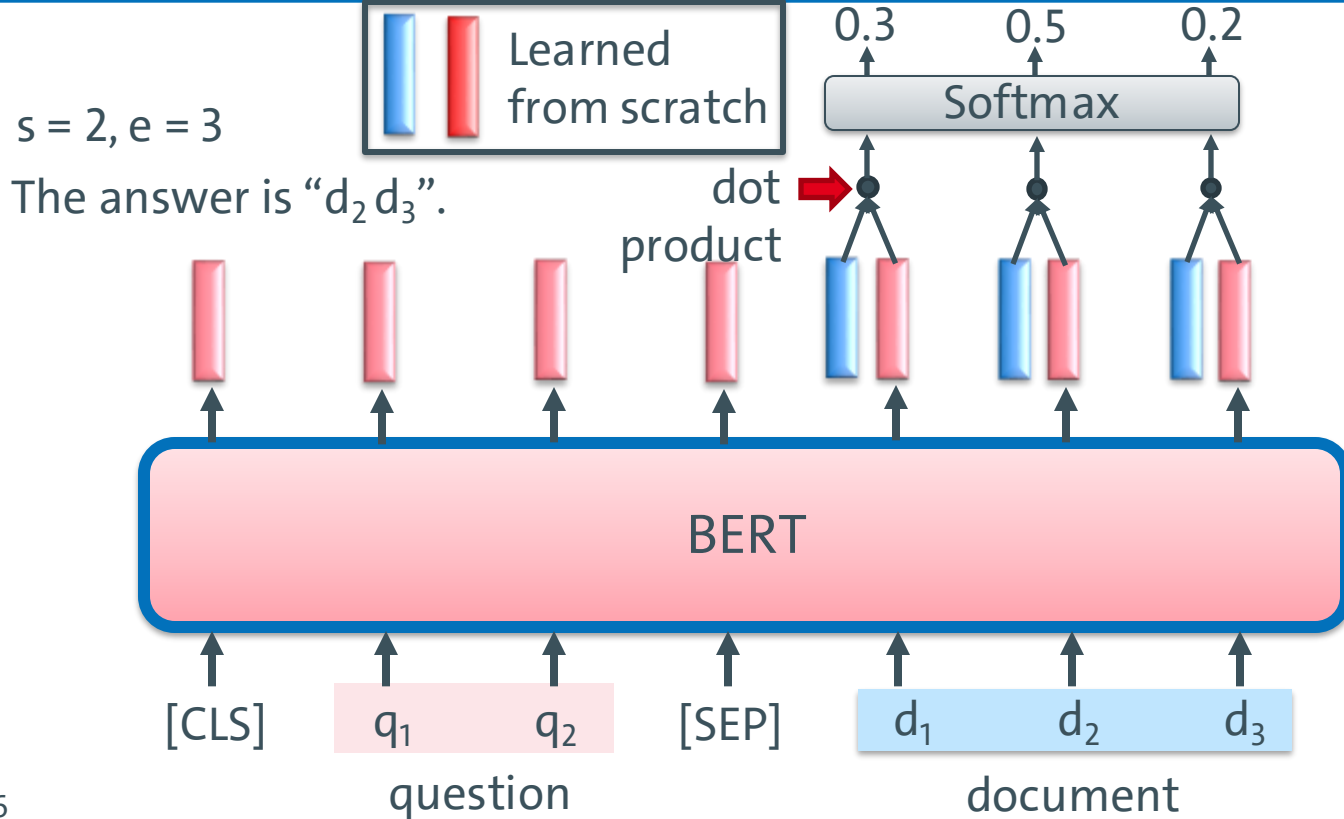
What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?

graupel

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

within a cloud s = 77, e = 79

How to use BERT – Case 4

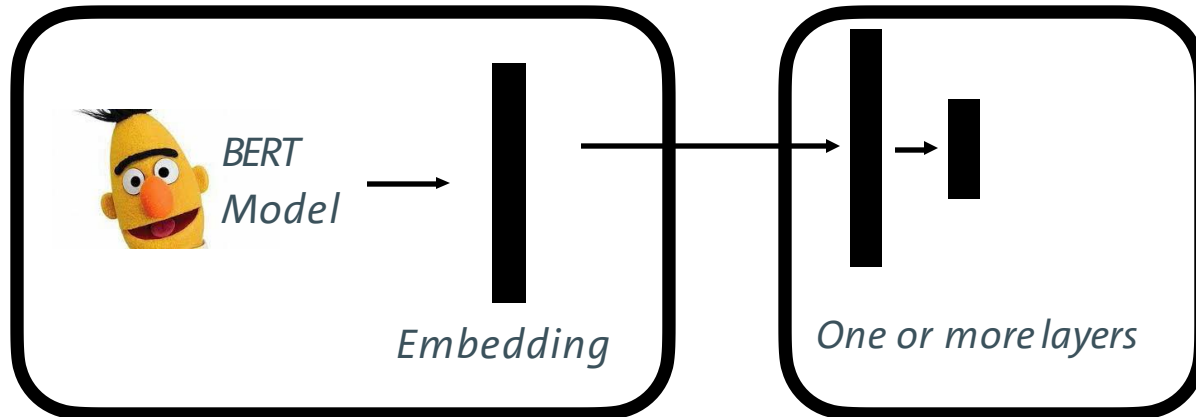


BERT Pre-Training & Fine Tuning

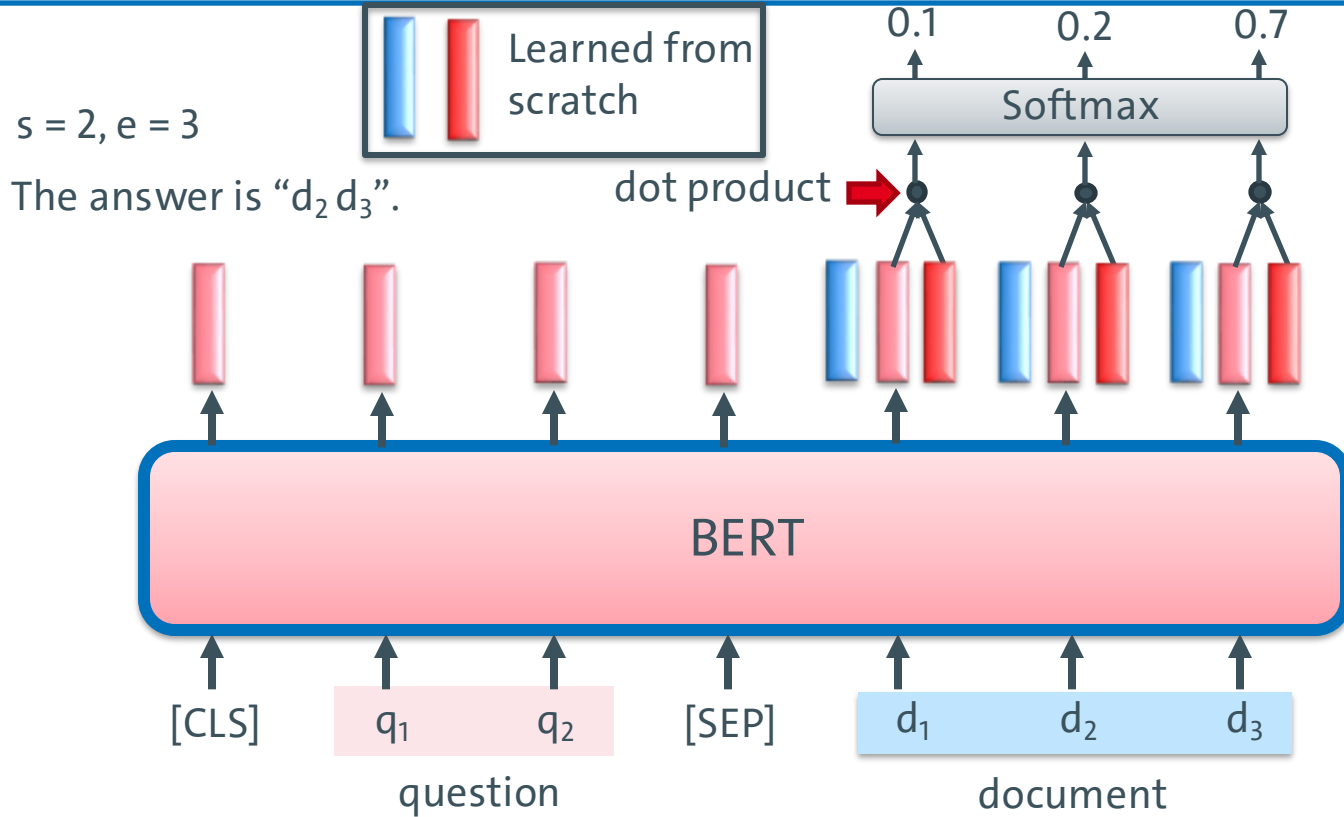
- Keep BERT frozen after pre-training
- Create BERT embeddings for labeled dataset for "downstream task" and train new model on these embeddings

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

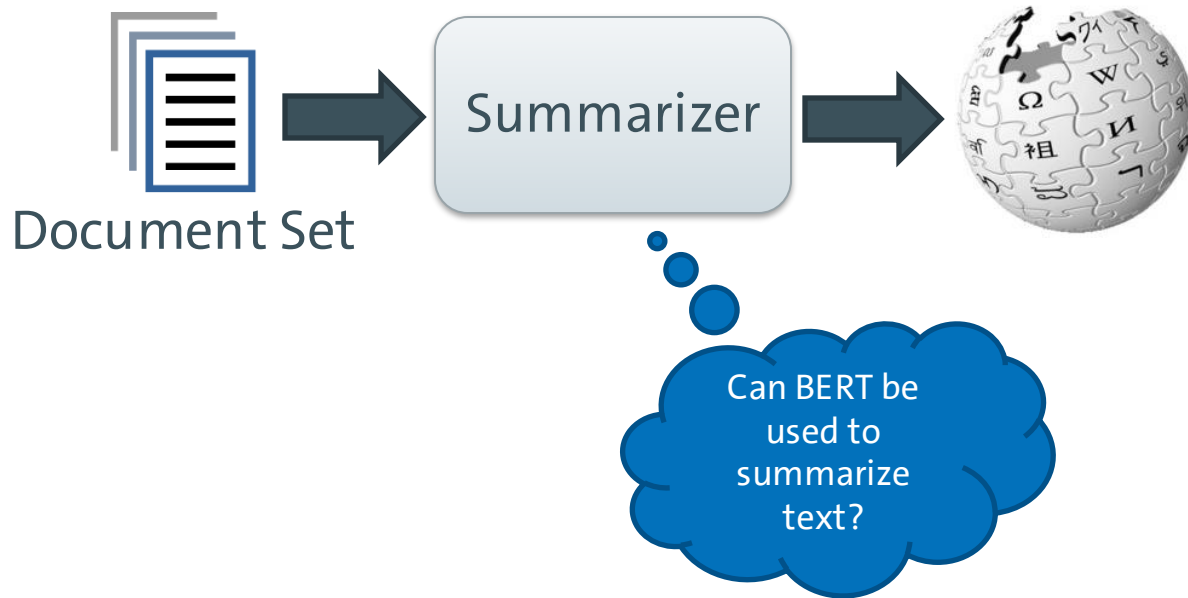
2) Feature-based training ("fine-tuning") on target task (supervised learning)



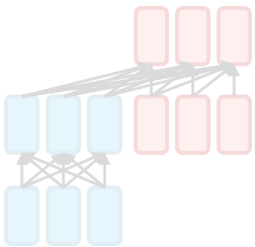
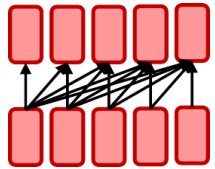
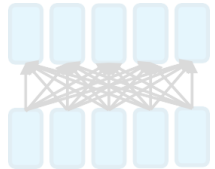
How to use BERT – Case 4



Example Application: Summarization



Model Pre-Training



- Encoder
 - Bidirectional context
 - Examples: BERT and its variants
- Decoder
 - Language modeling; better for generation
 - Example: GPT-2, GPT-3, LLama, [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.

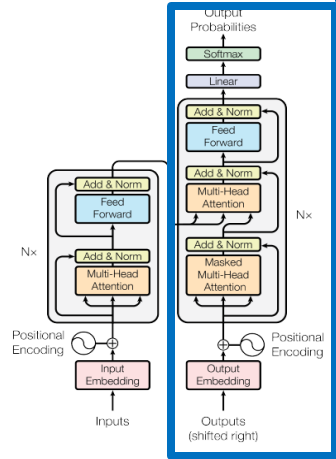
https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

GPT-3 (175 billion parameters)

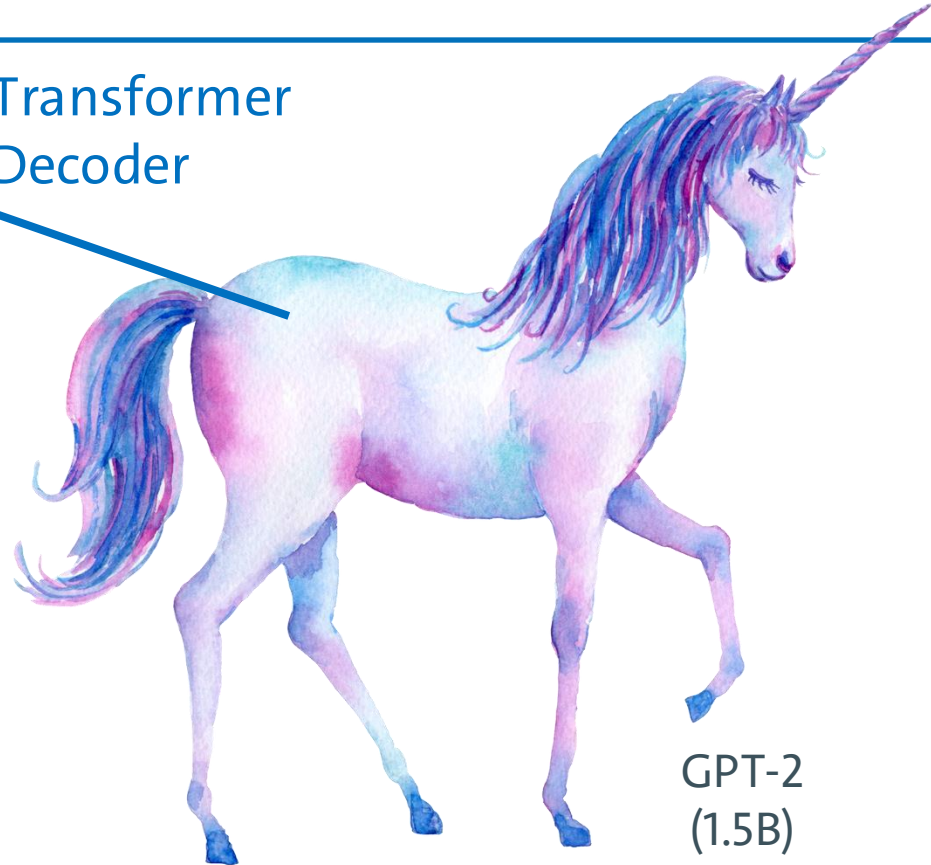
Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. arXiv preprint arXiv:2005.14165. <https://arxiv.org/abs/2005.14165>

GPT

<https://openai.com/index/better-language-models/>



Transformer Decoder



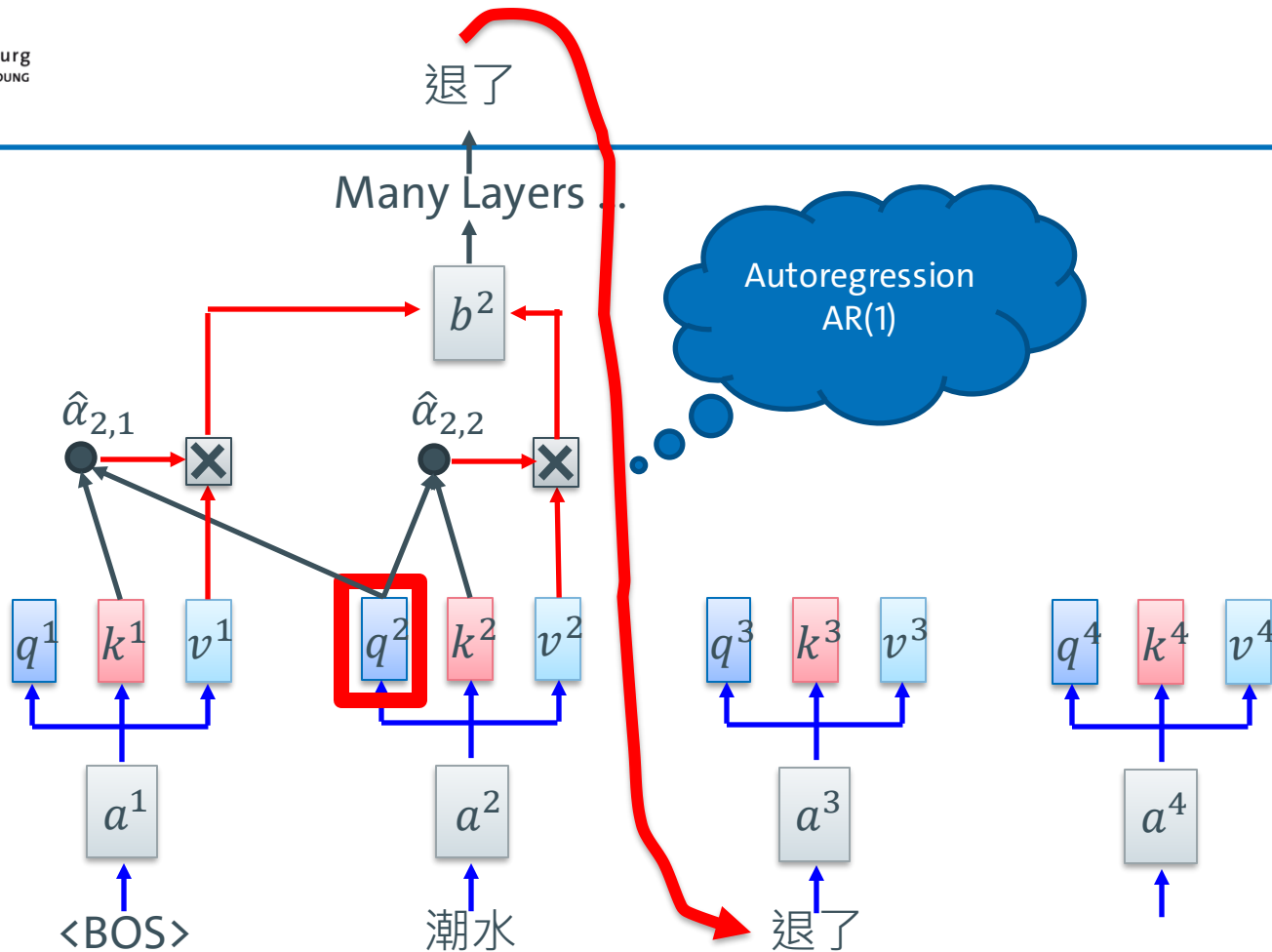
BERT
(340M)

ELMO
(94M)



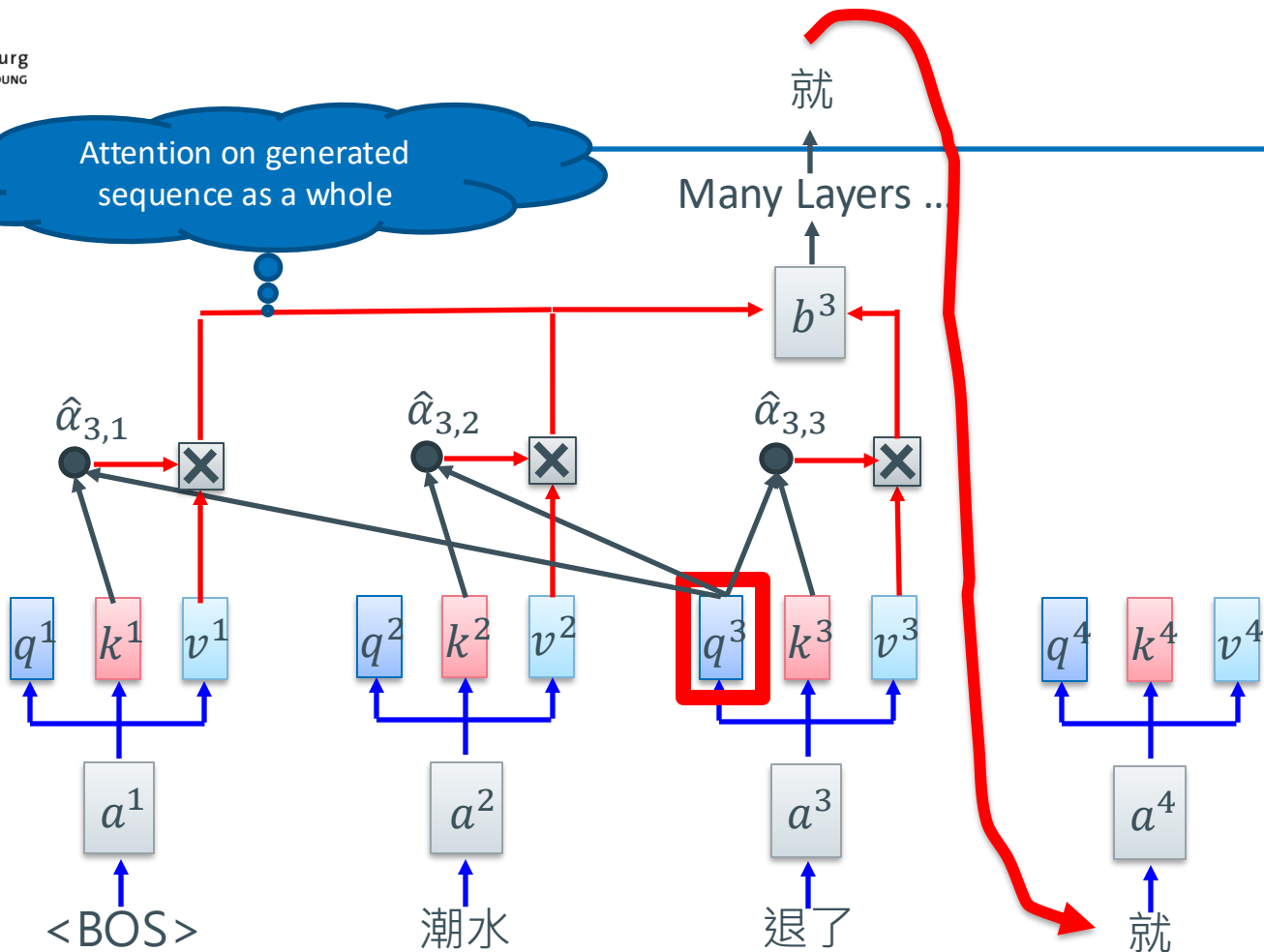
GPT-2
(1.5B)

GPT



GPT

Attention on generated sequence as a whole



Application: Summaries (Open AI)

The screenshot displays a digital application interface. On the left, there is a book cover for 'Alice's Adventures in Wonderland' by Lewis Carroll, featuring a blue background, a white rabbit, and a girl in a blue dress. To the right of the cover is a summary of the text. The summary is presented in a clean, white font on a dark background, with a progress bar at the bottom indicating the current position in the text. The summary includes the title, author, and the beginning of the first chapter, 'Down the Rabbit-Hole'.

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.

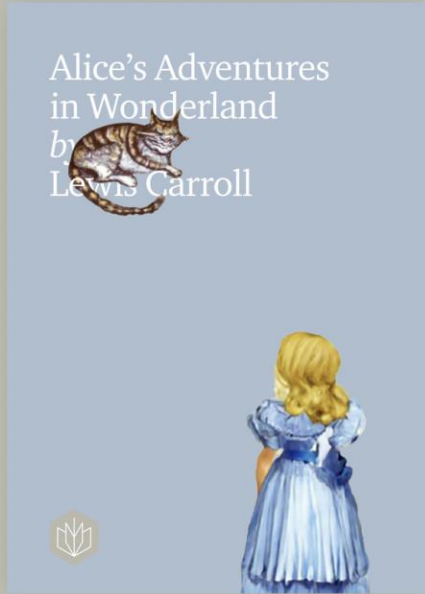
In another moment down went Alice after it, never once considering how to get out.

thousand miles down, I think-' (for several times of this sort in her k and though this was not a VEE showing off her knowledge, as the her, still it was good practice to say the right distance-but then I w Longitude I've got to?' (Alice had n or Longitude either, but thought th to say.)

Presently she began again. 'I w THROUGH the earth! How funny among the people that walk with il Antipathies, I think-' (she was rath listening, this time, as he didn't sour but I shall have to ask them what if you know. Please, Mr. N. at this N (and she tried to curtsy as she spo you're falling through the air! Do ye it?) 'And what an ignorant little gid No, it'll never do to ask: perhaps somewhere.

Down, down, down. There was r soon began talking again. 'Dinah! night, I should think!' (Dinah wa remember her saucer of milk at it wish you were down here with me! air, I'm afraid, but you might catch mouse, you know. But do cats eat Alice began to get rather sleep; herself, in a dreamy sort of way, 'D hats?' and complained 'The hole

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, and she wonders what latitude or longitude she'll fall right through the earth and where she'll emerge in is. She starts to get nervous about walking hand in hand with her sister, and she's ever eaten a bat. Suddenly she's in a moment.



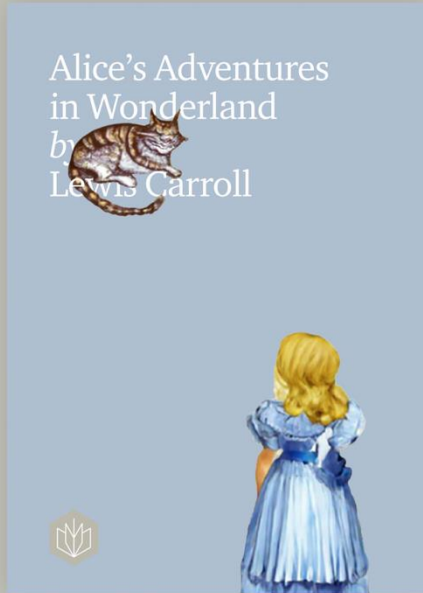
Section summaries are summarized again into higher-level summaries.

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 decides to go down the rabbit hole. She hears the White Rabbit and realizes she's now much bigger than the White Rabbit. She goes back to her normal size, and her second bottle makes her grow way into the garden. She doesn't like the Caterpillar, however. She sees the King and Queen and asks him to hookah on a mushroom and asks him to go back to her normal size. The Caterpillar says she'll be the same size as the mushroom will make her grow. She starts eating the mushroom and 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.