Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 5.2: Alignment and Representation

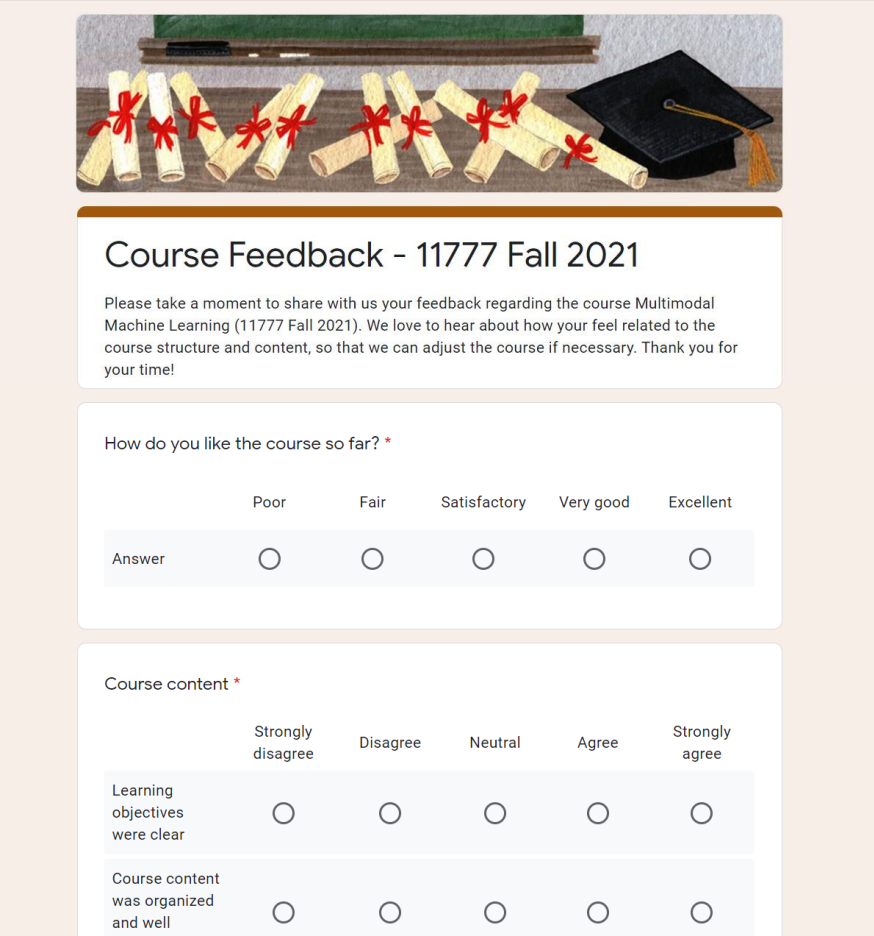
Louis-Philippe Morency

** Original course co-developed with Tadas Baltrusaitis.*
Spring 2021 edition taught by Yonatan Bisk

Administrative Stuff

Share Your Thoughts!

https://forms.gle/ZUBcMZVf4Ttv2uQ66



Deadline

Please submit your feedback about this course before this Sunday 10/3

Optional,
but greatly appreciated! ☺

Anonymous, by default.

- You can optionally share your email address if you want us to follow-up with you directly.

Language Technologies Institute

Carnegie Mellon University

Multimodal Machine Learning

Lecture 5.2: Alignment and Representation

Louis-Philippe Morency

** Original course co-developed with Tadas Baltrusaitis.*
Spring 2021 edition taught by Yonatan Bisk

Objectives of today's class

- Soft attention models
- Contextualized sentence embedding
- Transformer networks
  - Self-attention
  - Multi-head attention
  - Position embeddings
  - Sequence-to-sequence modeling
- Multimodal contextualized embeddings
- Language pre-training
  - BERT pre-training and fine-tuning

Implicit and “Uni-Directional” Alignment

Modality A
(query)

A woman is throwing a frisbee

Modality B
(key)

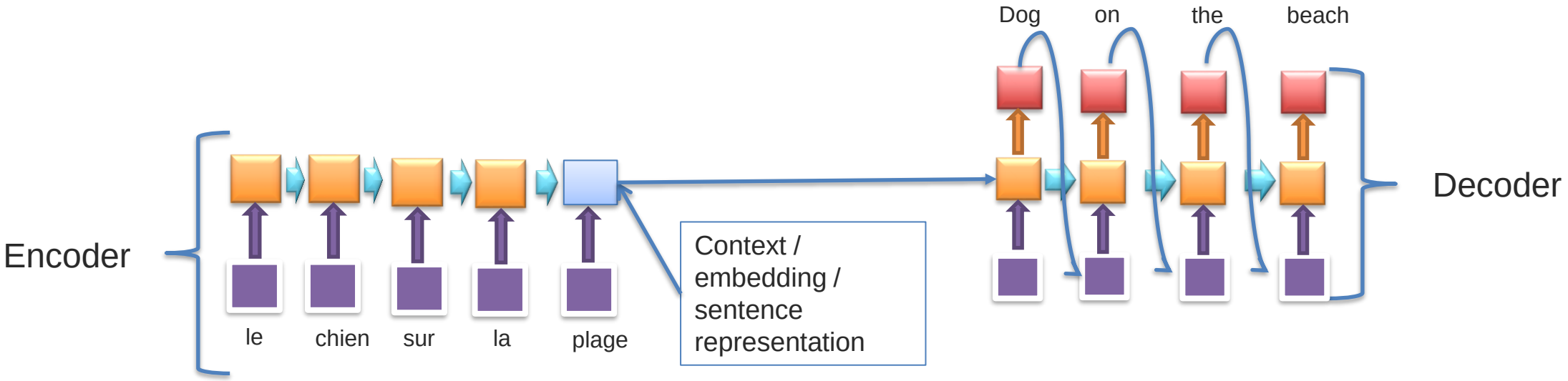
1. Hard attention

2. Warping

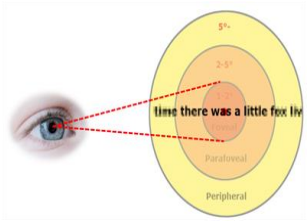
3. Soft attention
(discussed on today!)

Soft Attention Models

Sequence-to-Sequence Models



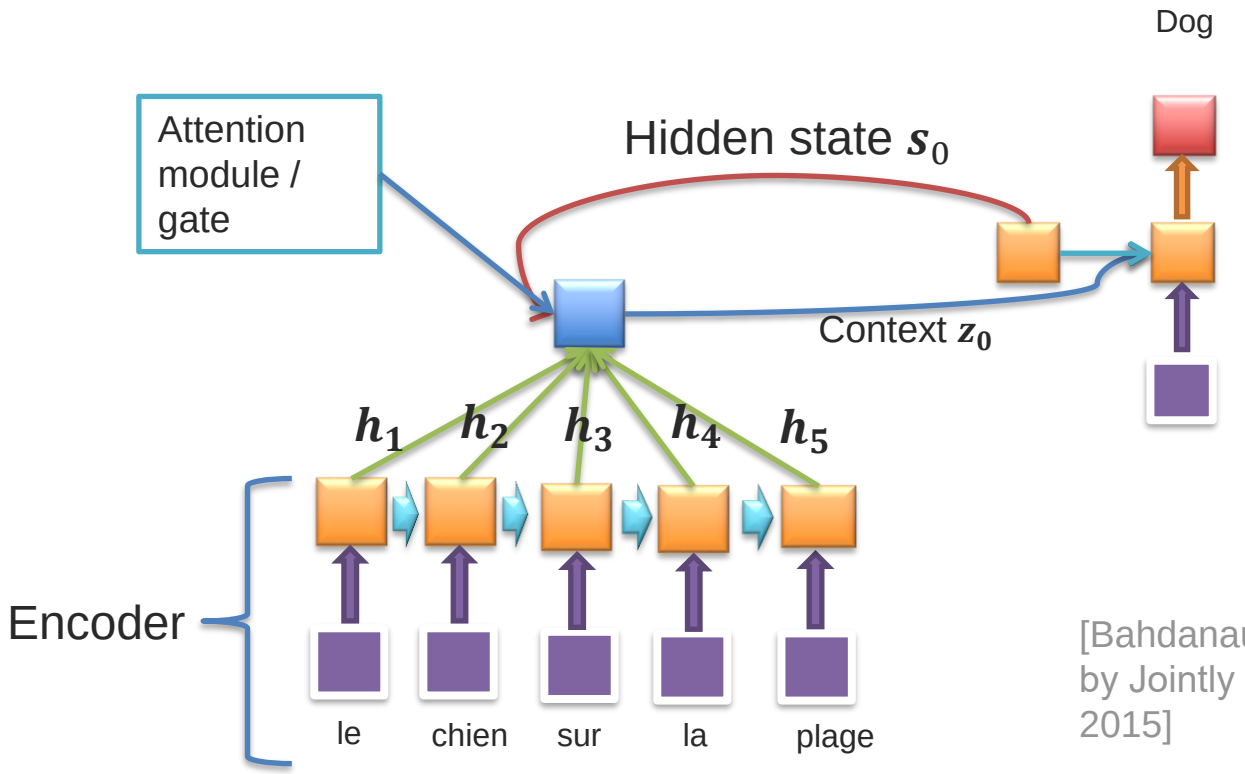
What is the problem with this?
What happens when the sentences are very long?



Inspiration:
human attention

Decoder with Attention Module

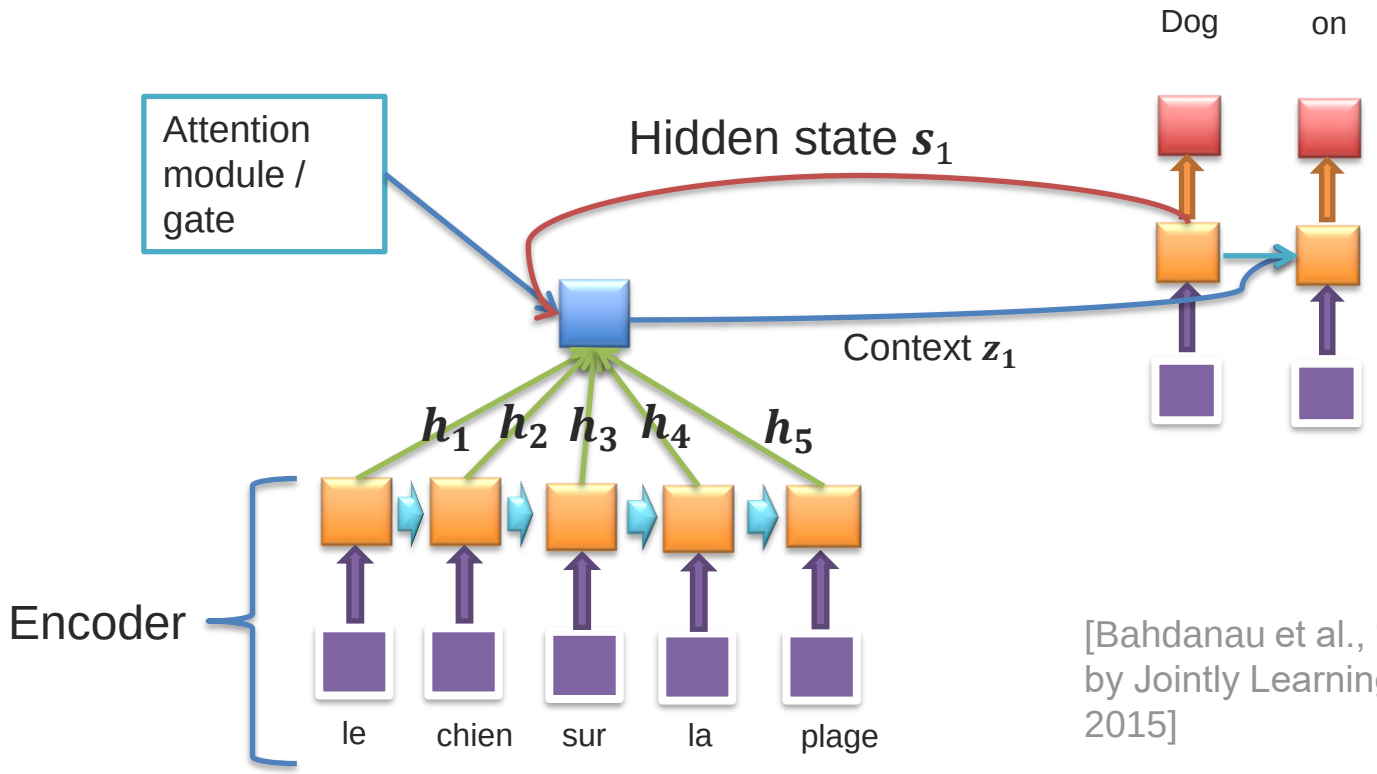
A new intermediate hidden state is computed for each decoding iteration:



[Bahdanau et al., "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015]

Decoder with Attention Module

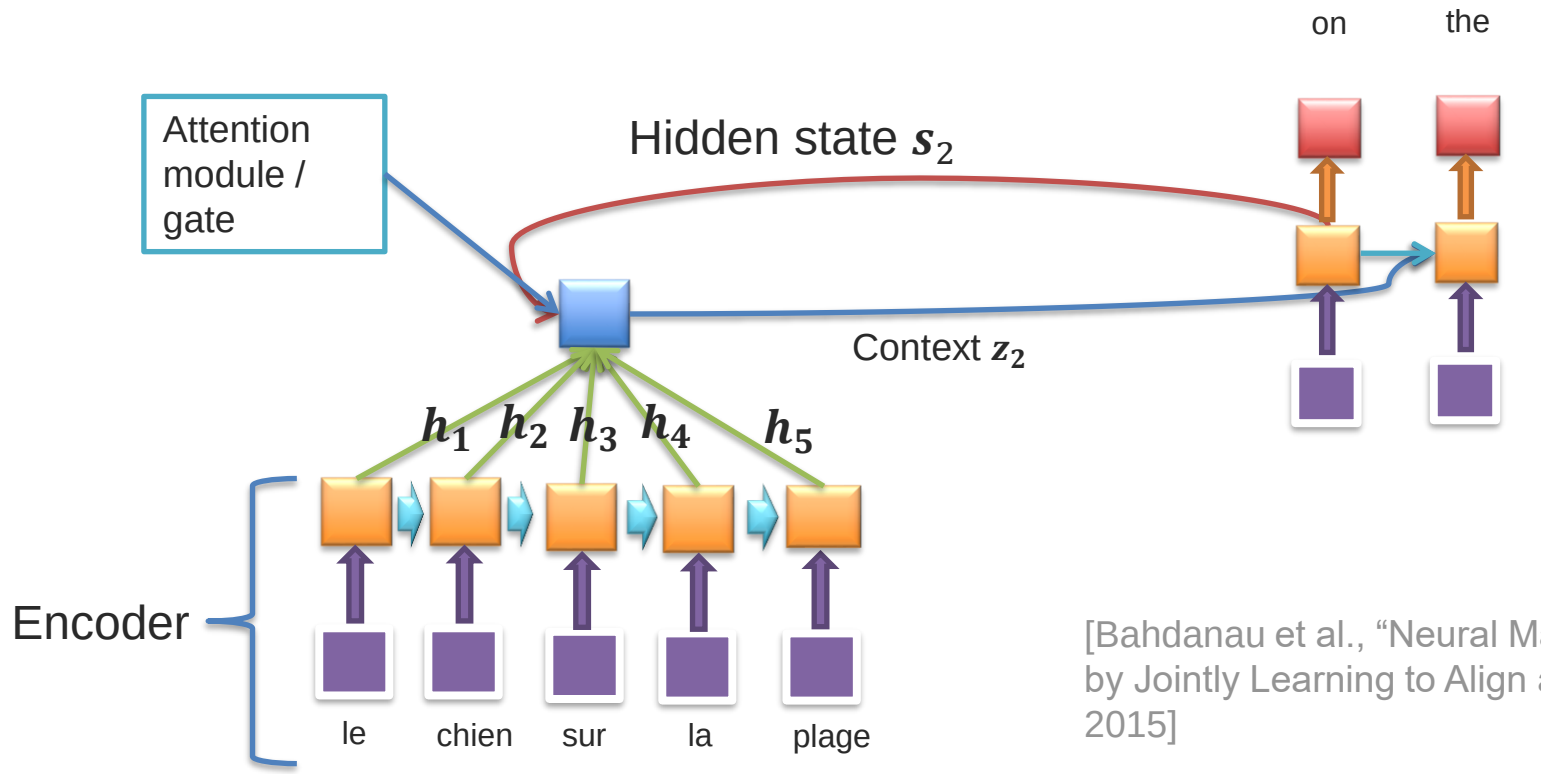
A new intermediate hidden state is computed for each decoding iteration



[Bahdanau et al., "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015]

Decoder with Attention Module

A new intermediate hidden state is computed for each decoding iteration



[Bahdanau et al., "Neural Machine Translation by Jointly Learning to Align and Translate", ICLR 2015]

How do we encode attention?

Before:

$$p(y_i|y_1, \dots, y_{i-1}, \boldsymbol{x}) = g(y_{i-1}, \boldsymbol{s}_i, \boldsymbol{z}),$$

where $\boldsymbol{z} = \boldsymbol{h}_T$, last encoder state and $\boldsymbol{s}_i$ is the current state of the decoder

Now:

$$p(y_i|y_1, \dots, y_{i-1}, \boldsymbol{x}) = g(y_{i-1}, \boldsymbol{s}_i, \boldsymbol{z}_i)$$

Have an attention "gate"

- A different context $\boldsymbol{z}_i$ used at each time step!
- $\boldsymbol{z}_i = \sum_{j=i}^{T_x} \alpha_{ij} \boldsymbol{h}_j$

α_{ij} is the (scalar) attention for word j at generation step i

How do we encode attention?

So how do we determine α_{ij}?

$\alpha_{i,j} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$ - softmax, making sure they sum to 1

where:

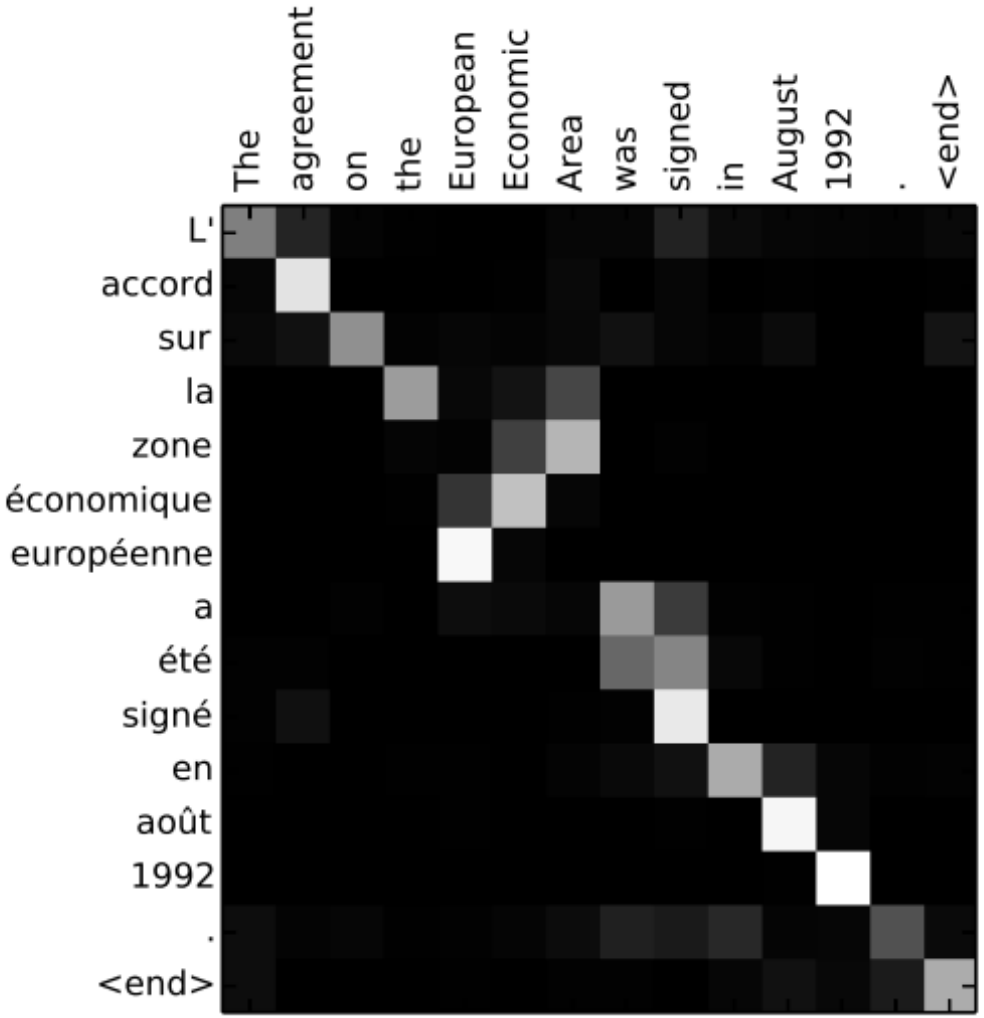
$e_{ij} = \boldsymbol{v}^T \sigma\left(W s_{i-1} + U h_j\right)$

a feedforward network that can tell us how important the current encoding is

$\boldsymbol{v}, W, U$– learnable weights

$z_i = \sum_{j=i}^{T_x} \alpha_{ij} h_j$ ← expectation of the context (a fancy way to say it's a weighted average)

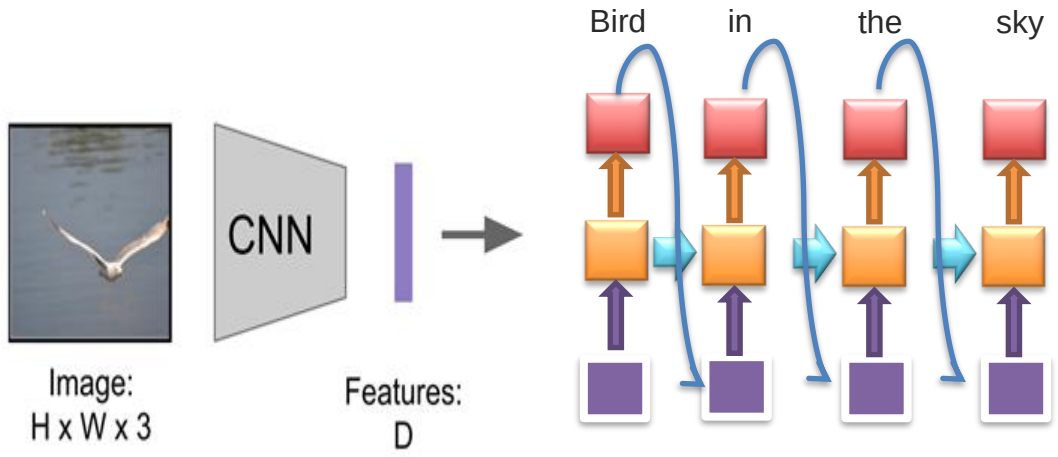
Example – Attention for Machine Translation



Example – Visual captioning

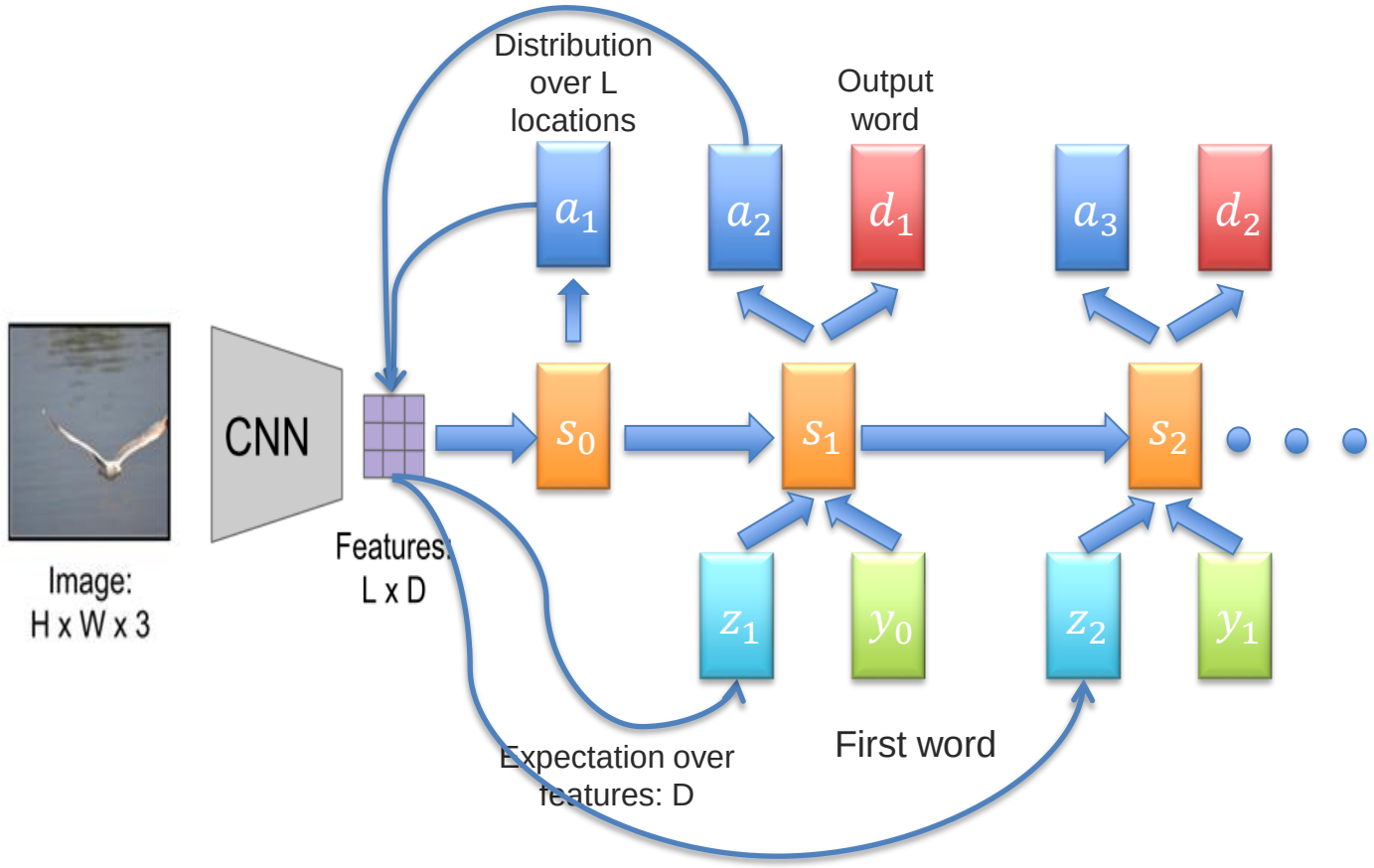
[Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, Xu et al., 2015]

Recap RNN for Captioning



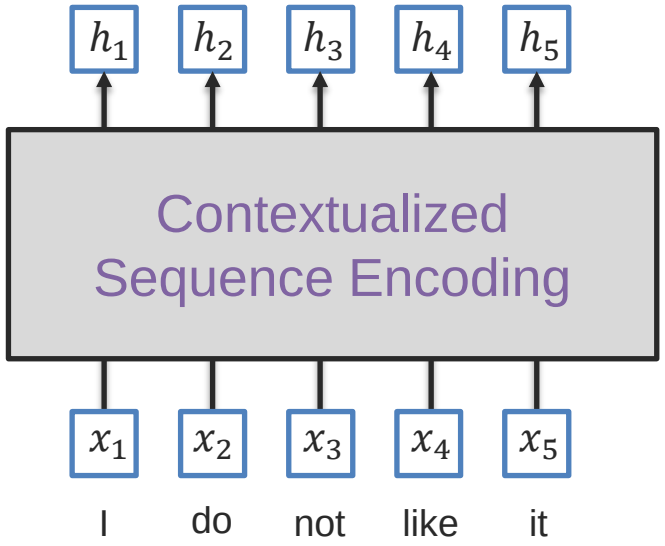
Why not using final layer of the CNN?

Looking at more fine grained features



Attention for (contextualized) Representation Learning

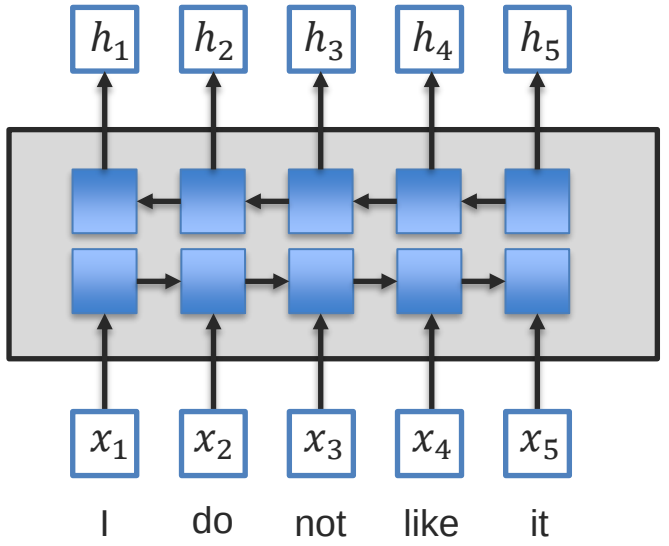
Sequence Encoding - Contextualization



How to encode this sequence while modeling the interaction between elements (e.g., words)?

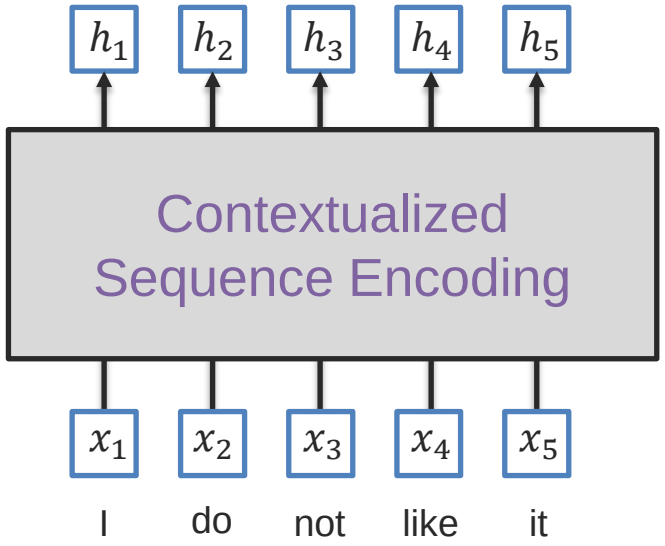
Option 1: Bi-directional LSTM:

(e.g., ELMO)

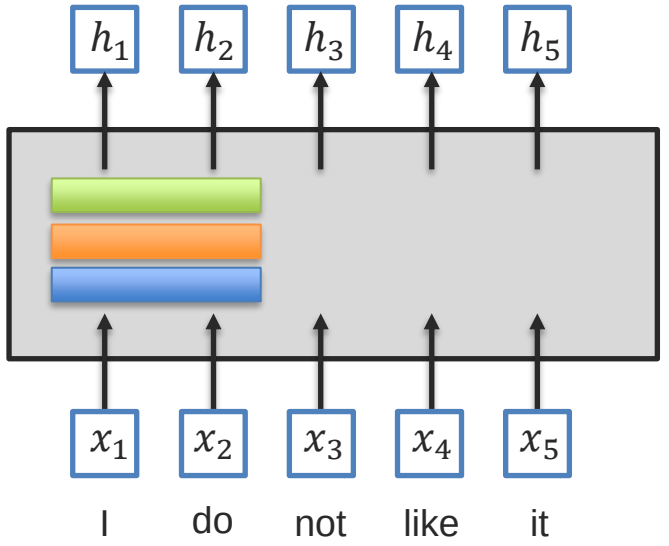


But harder to parallelize…

Sequence Encoding - Contextualization



Option 2: Convolutions

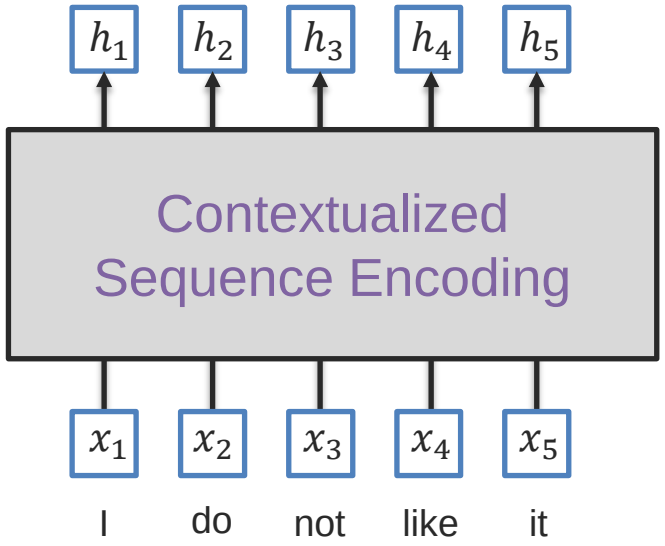


Can be parallelized!

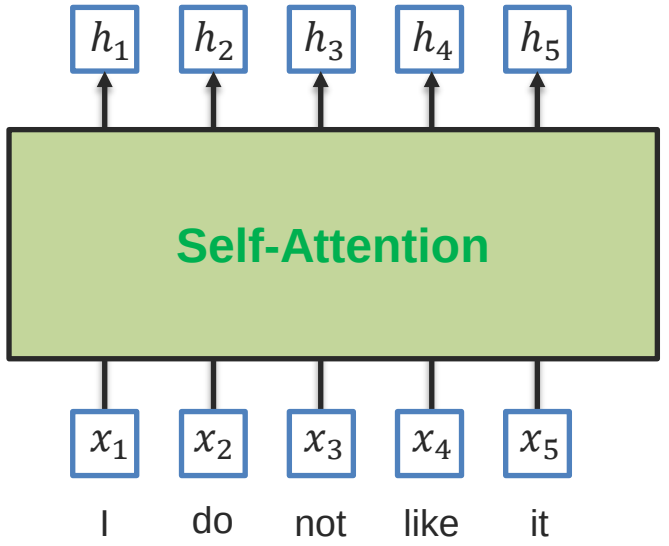
But modeling long-range dependencies require multiple layers

And convolutional kernels are static

Sequence Encoding - Contextualization



Option 3: Self-attention



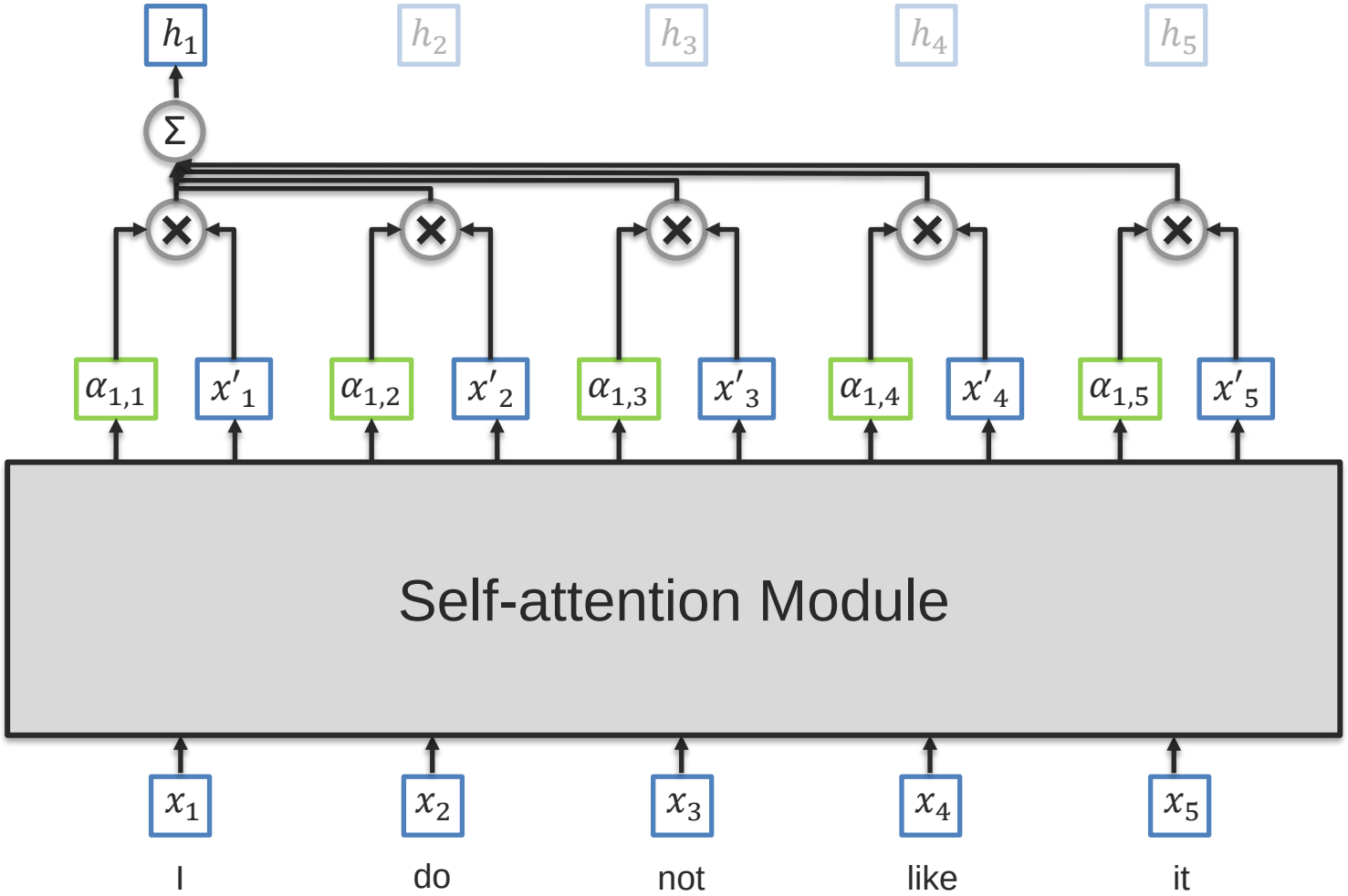
Can be parallelized!

Long-range dependencies

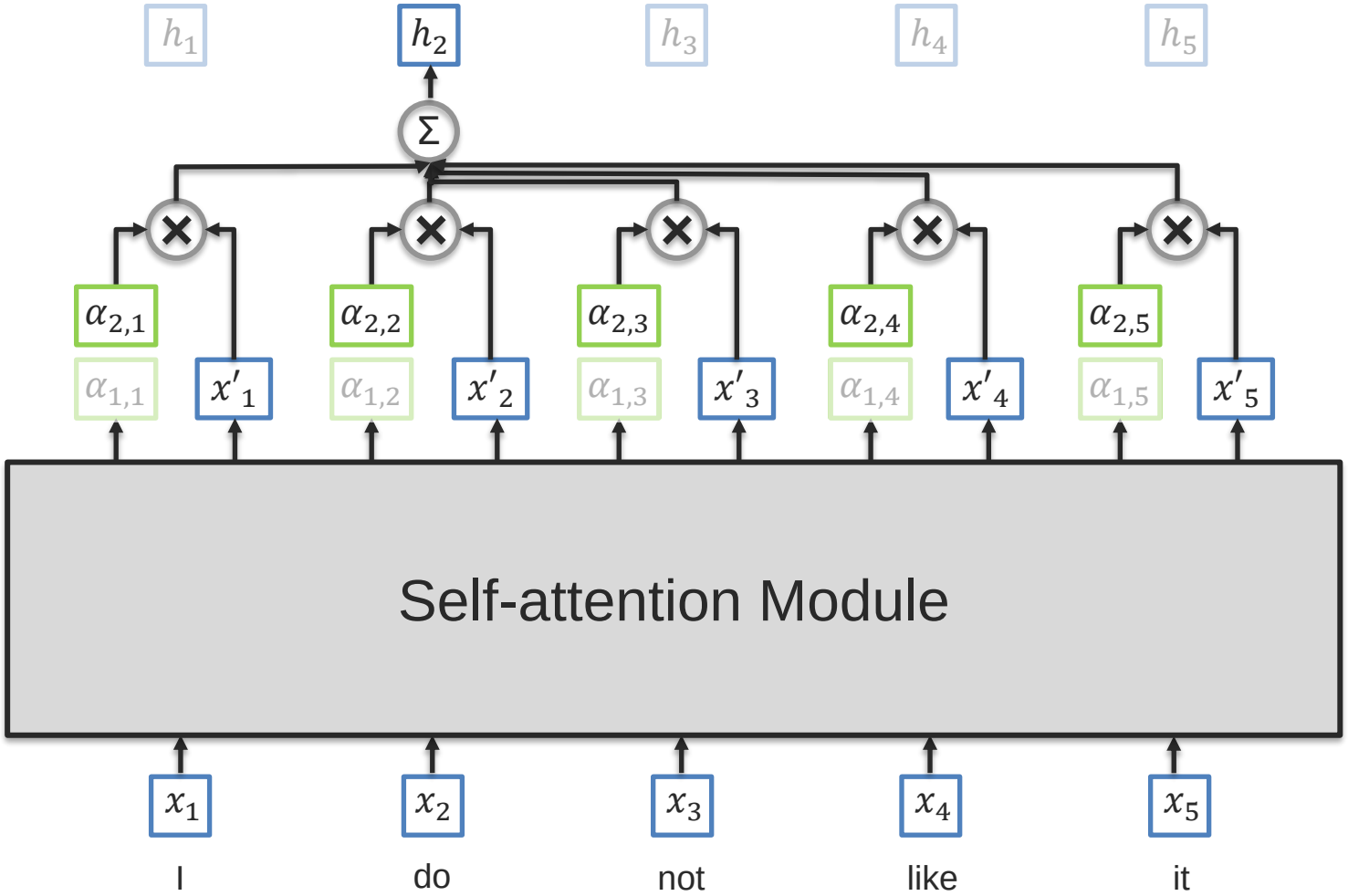
Dynamic attention weights

Self-Attention

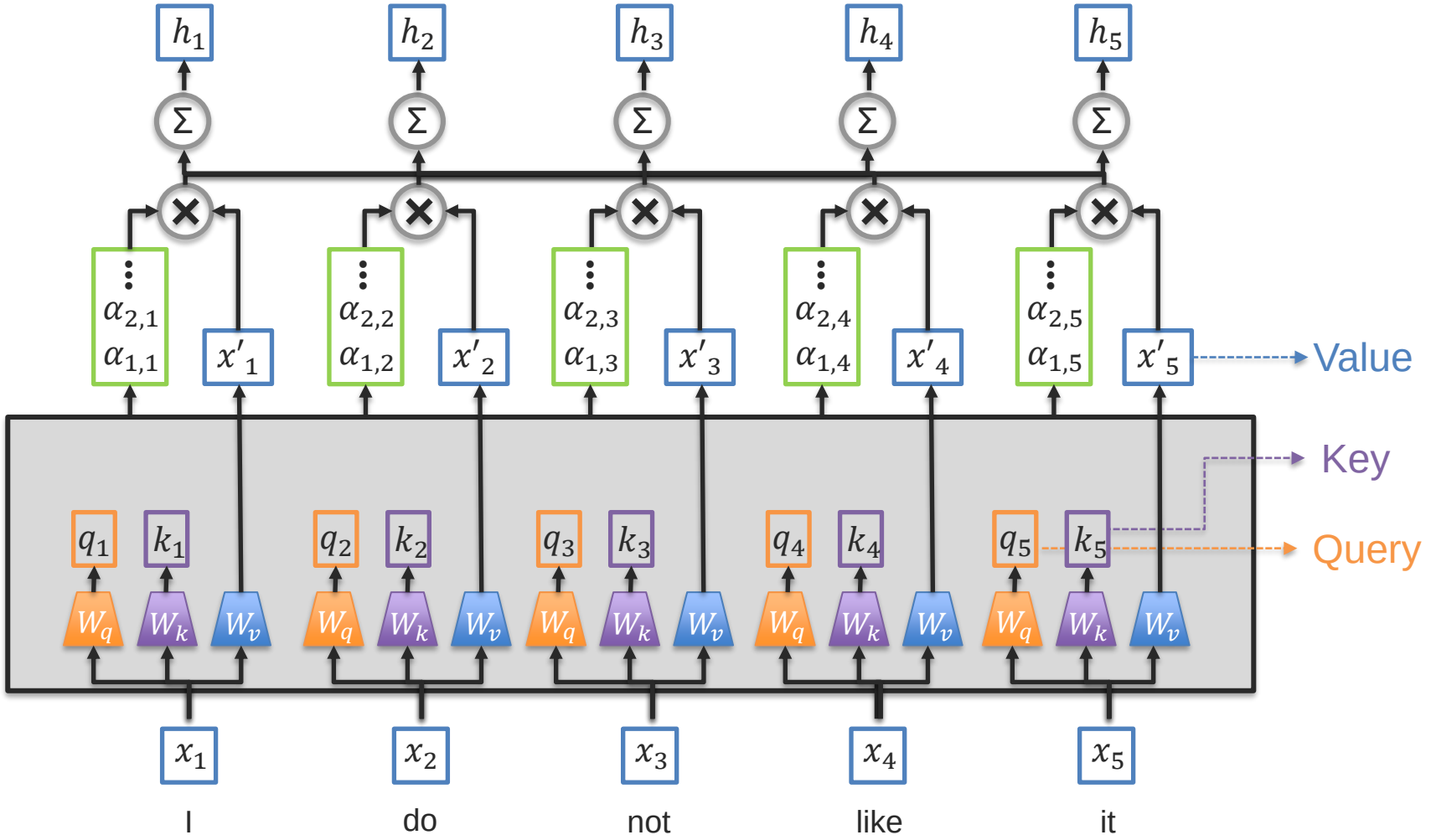
Self-Attention



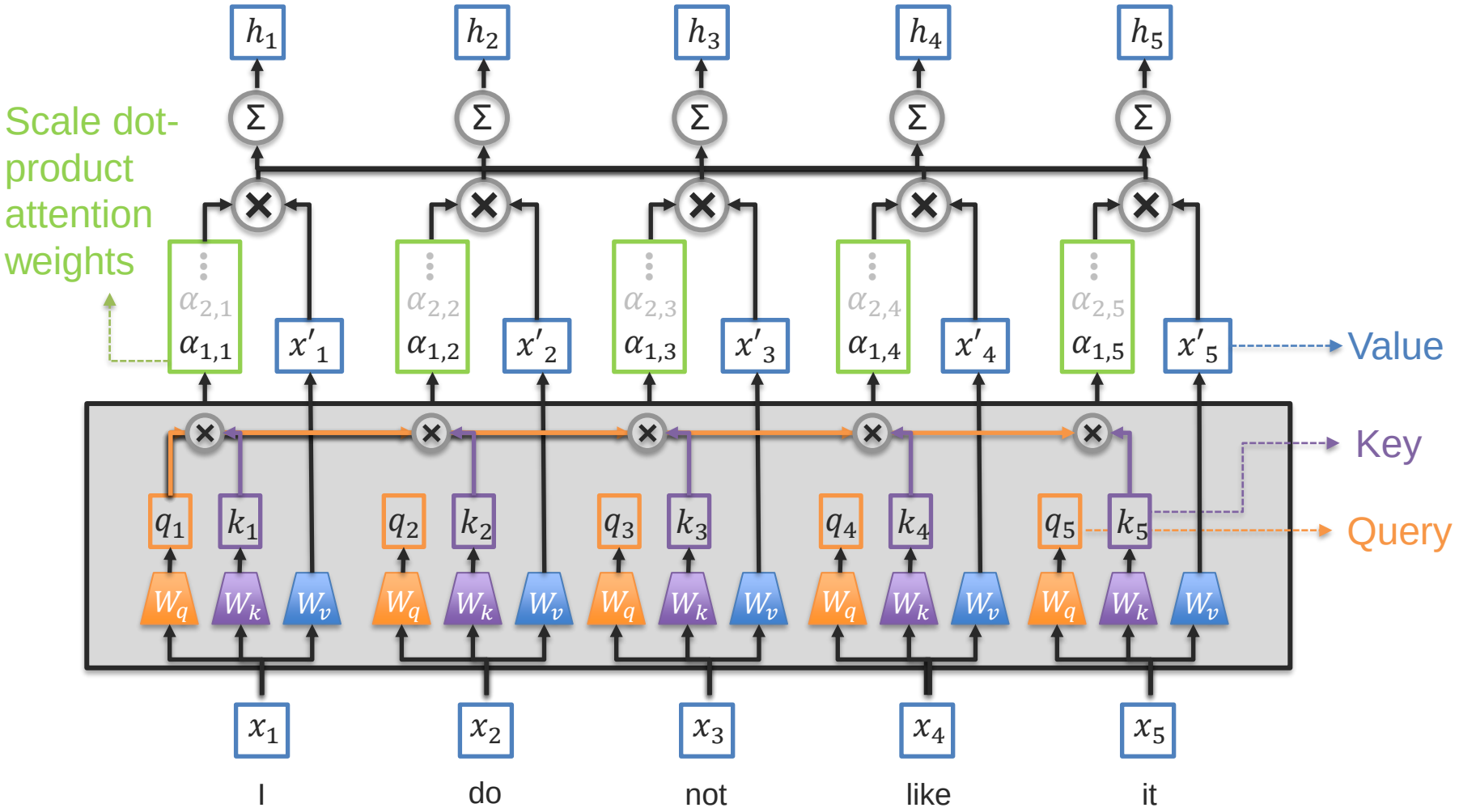
Self-Attention



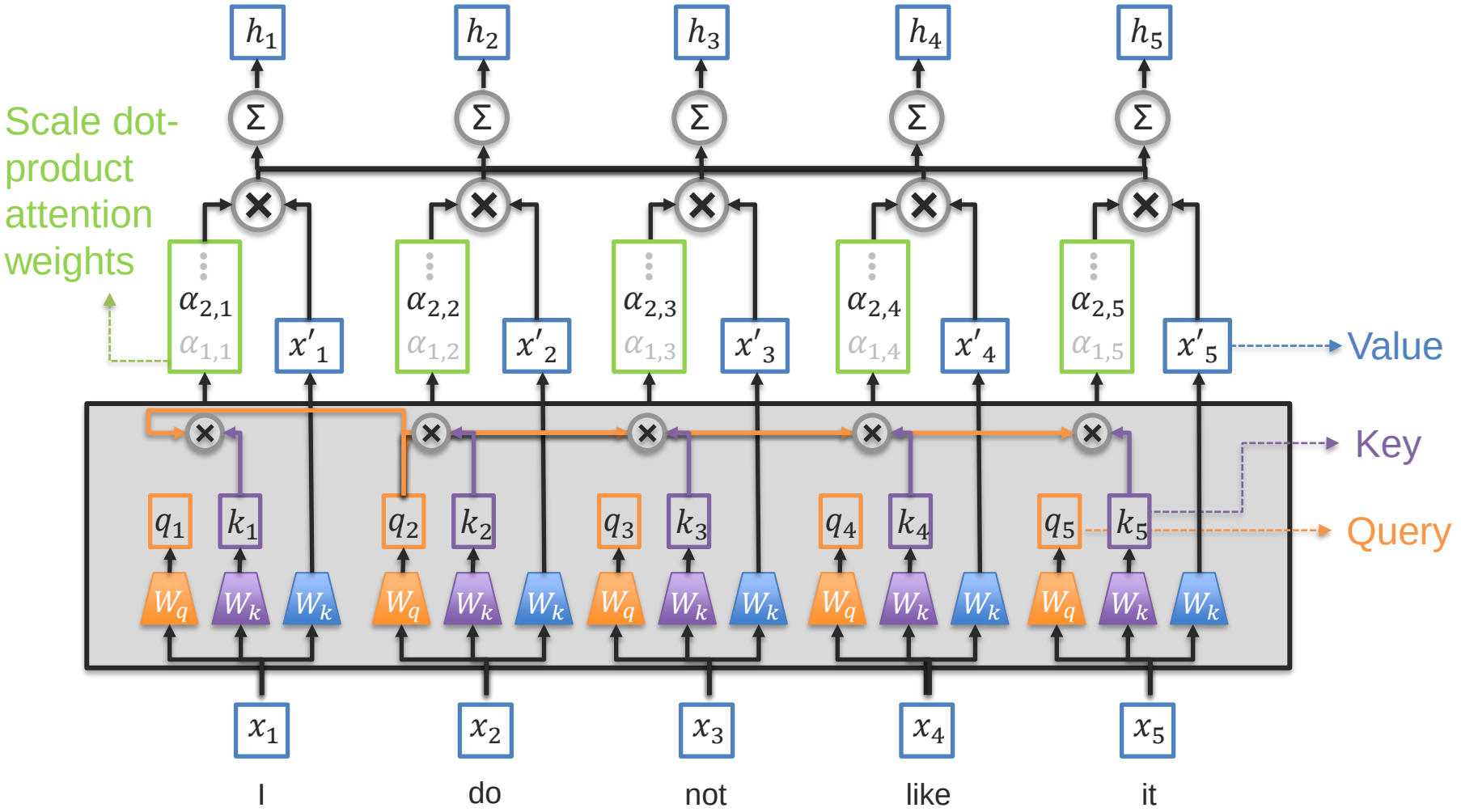
Transformer Self-Attention



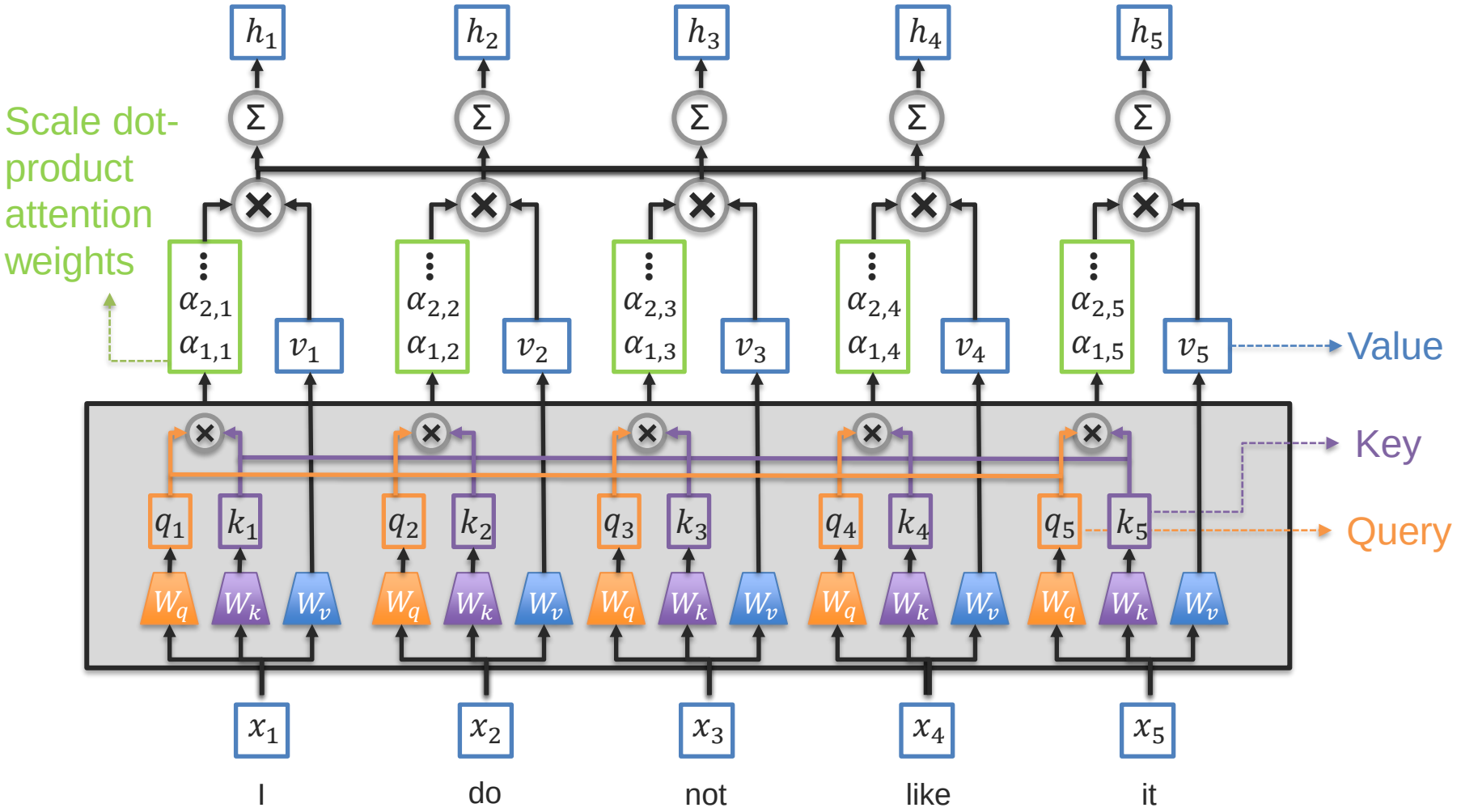
Transformer Self-Attention



Transformer Self-Attention

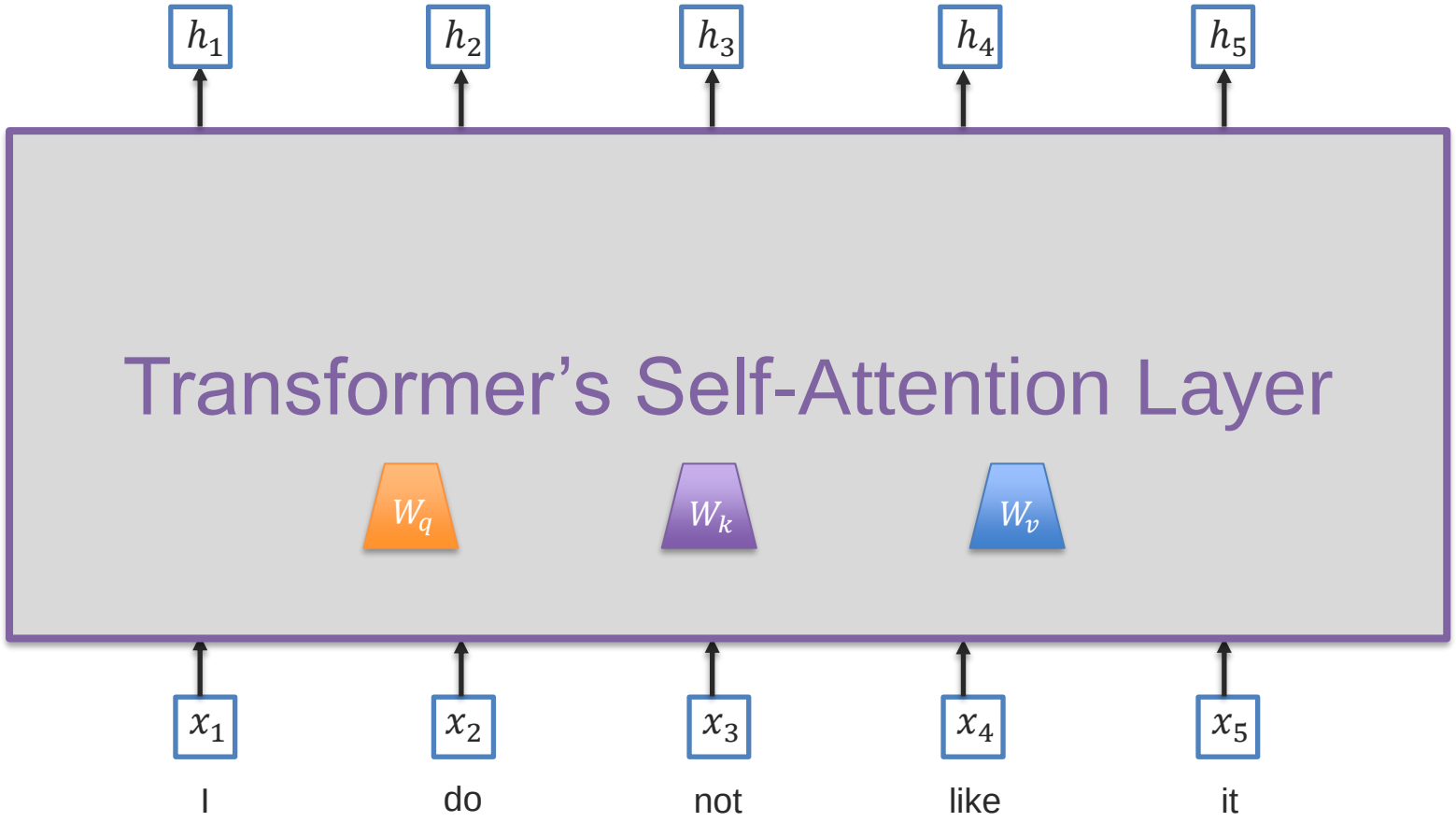


Transformer Self-Attention

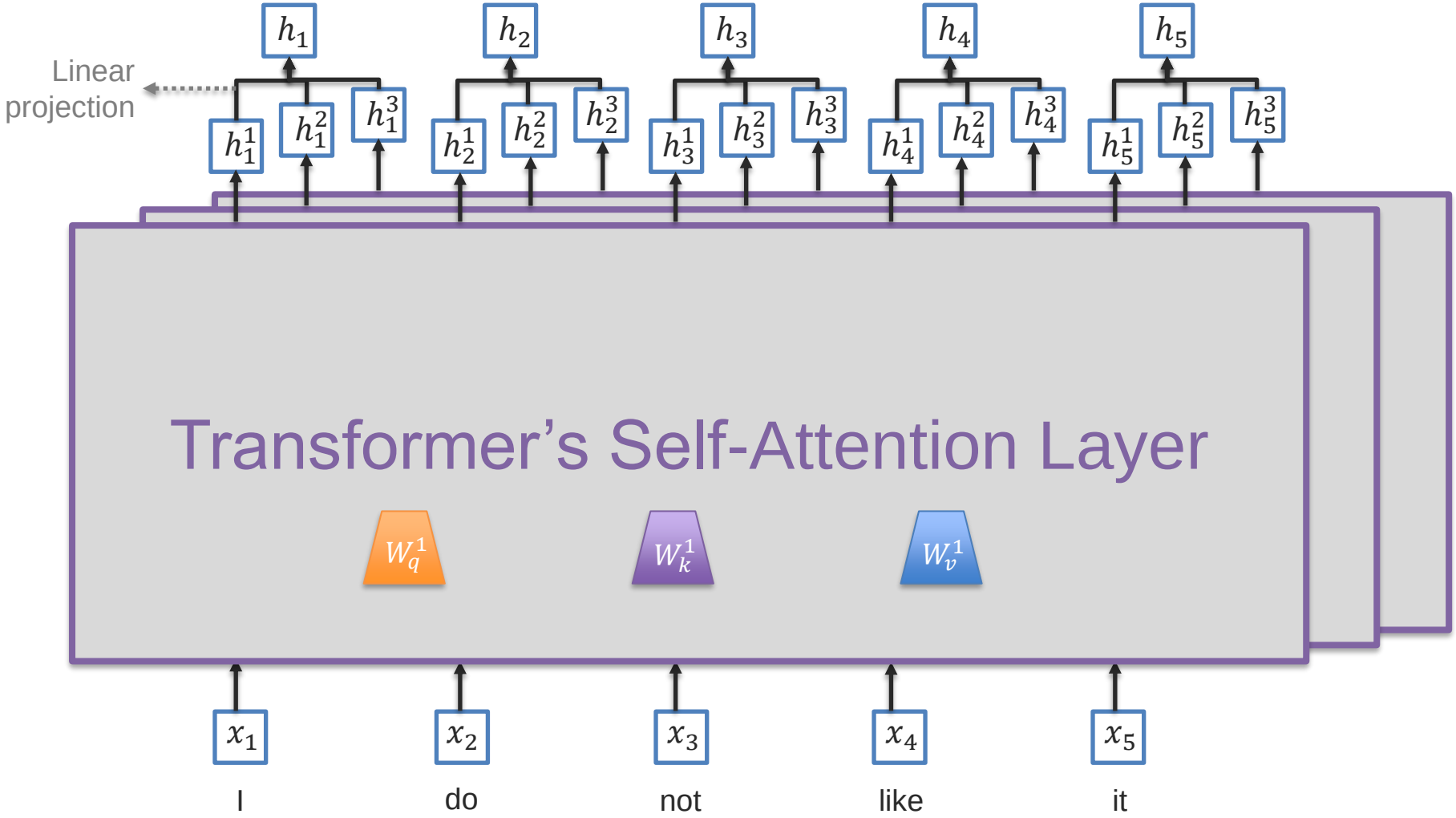


Transformer Self-Attention

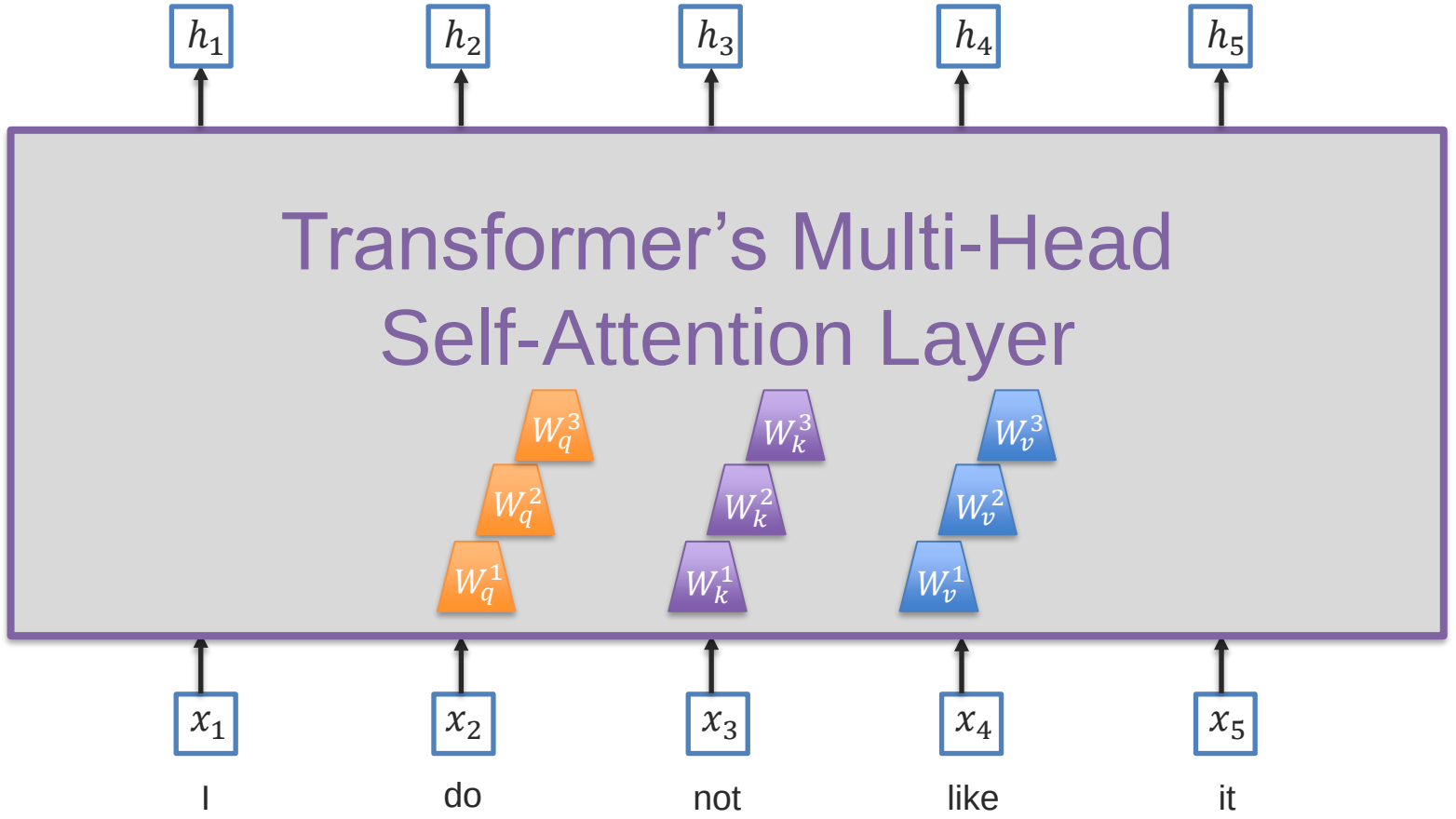
What if we want to attend simultaneously to multiple subspaces of x?



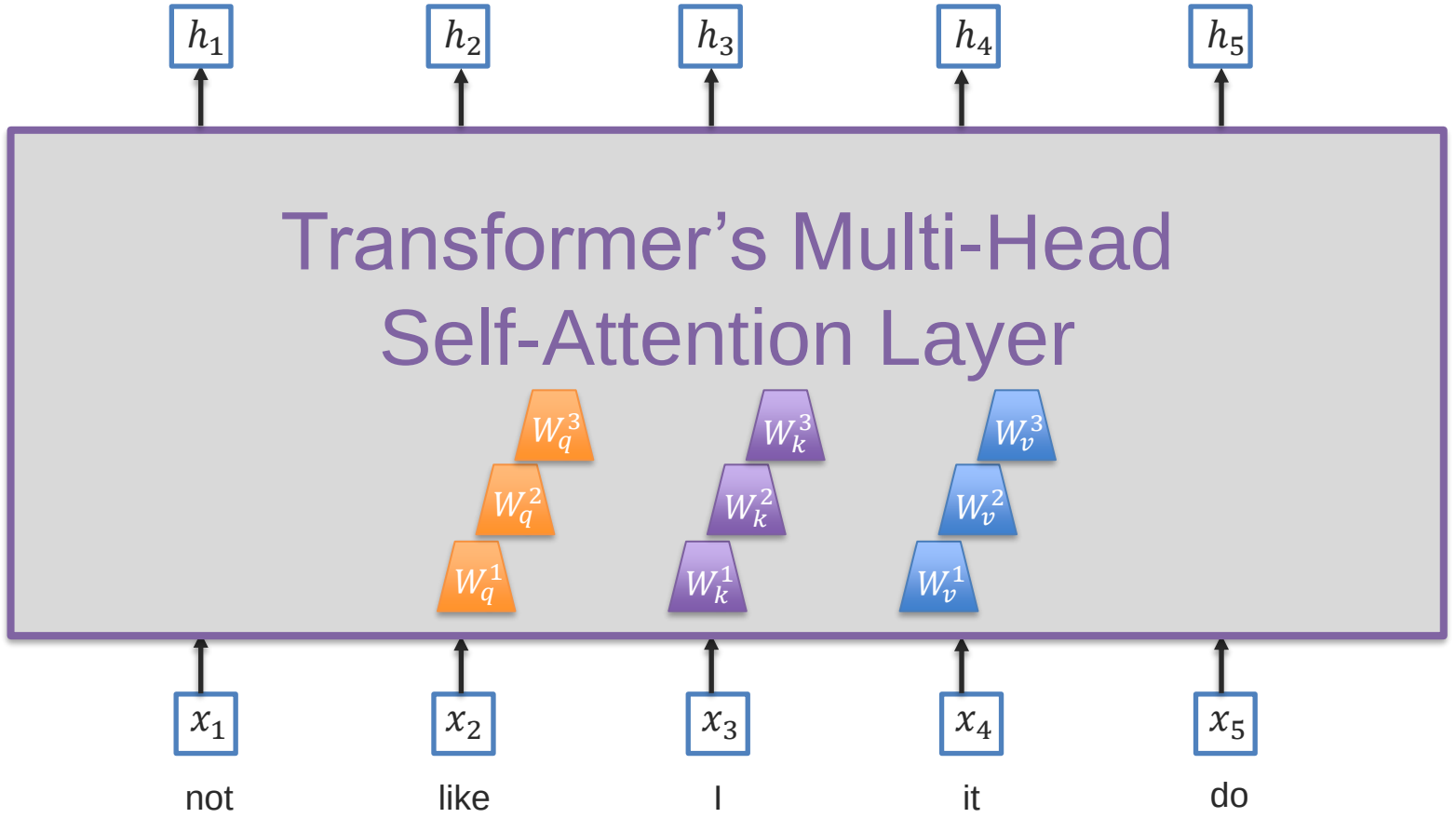
Transformer Multi-Head Self-Attention



Transformer Multi-Head Self-Attention



Transformer Multi-Head Self-Attention



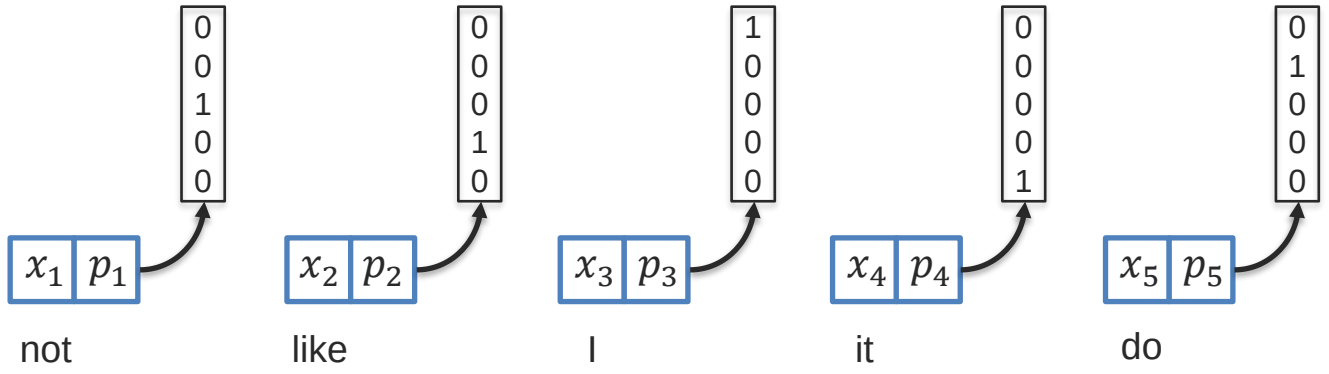
What happens if the words are shuffled?

Position embeddings

- Position information is not encoded in a self-attention module

How can we encode position information?

Simple approach: one-hot encoding

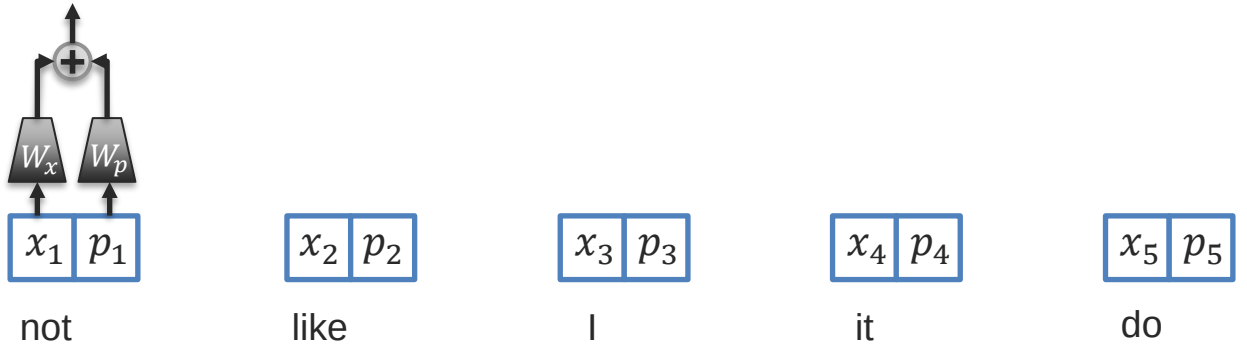


Position embeddings

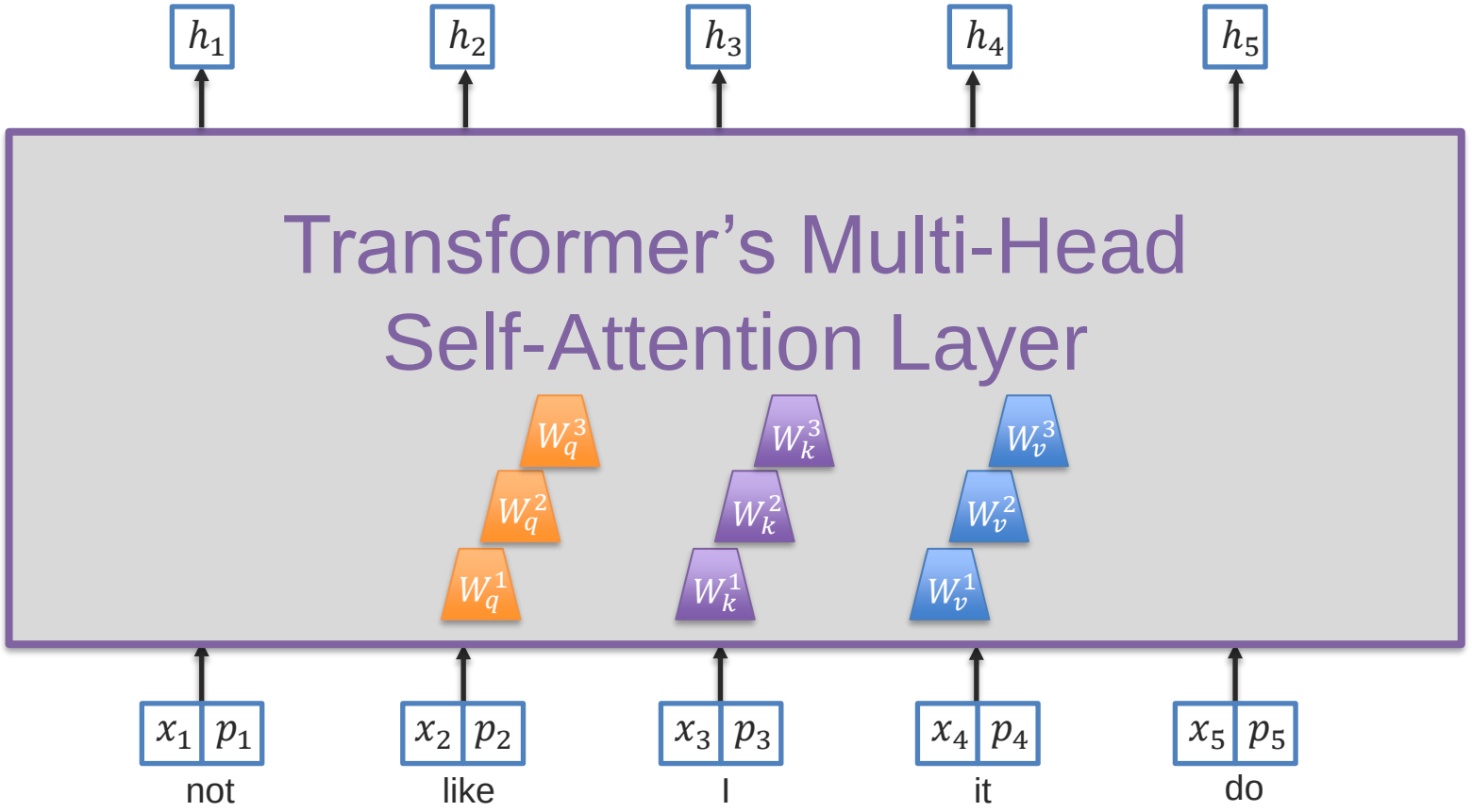
- Position information is not encoded in a self-attention module

How can we encode position information?

Simple approach: one-hot encoding + linear embeddings + { Sum - or - concat

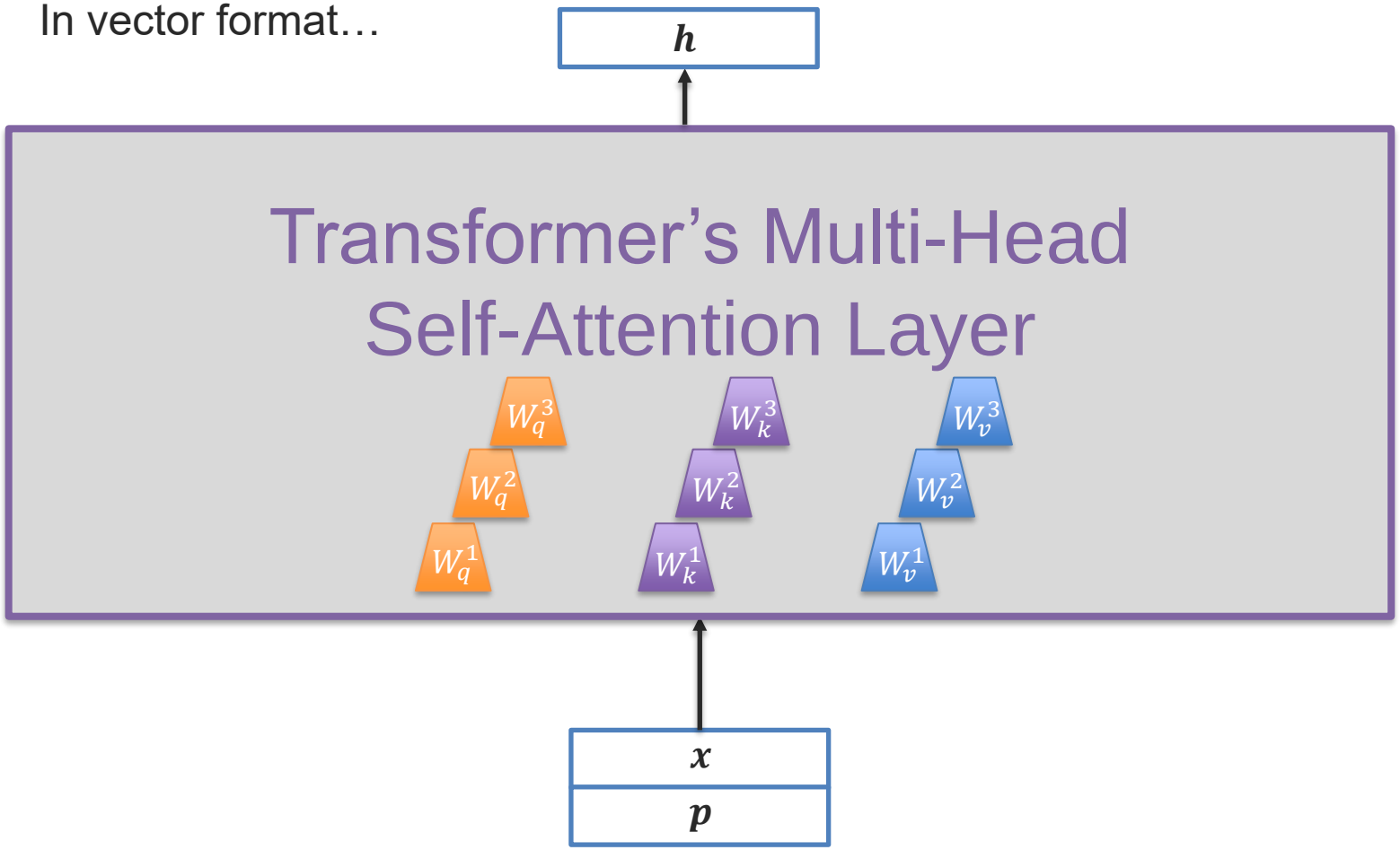


Transformer Multi-Head Self-Attention

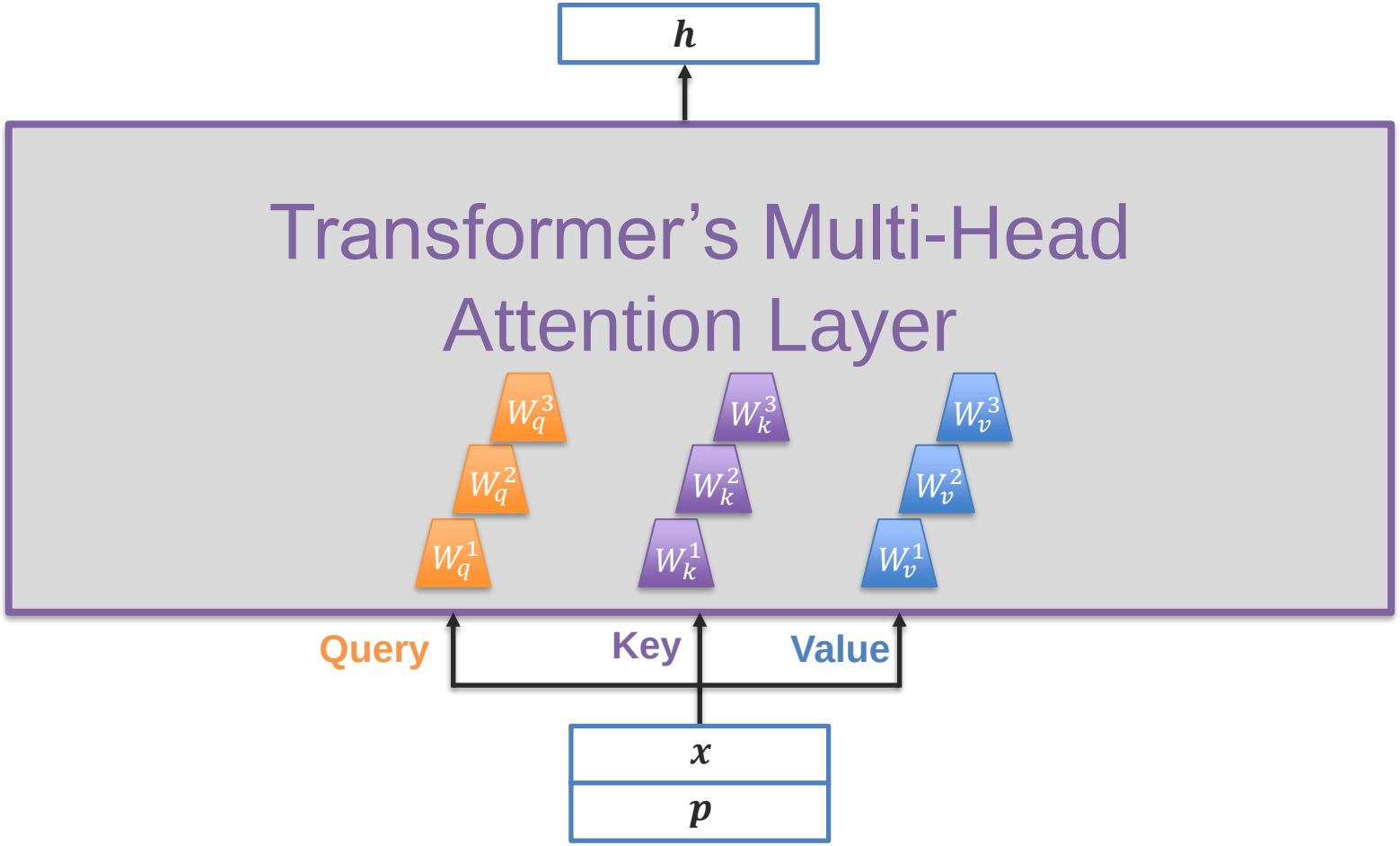


Transformer Multi-Head Self-Attention

In vector format…

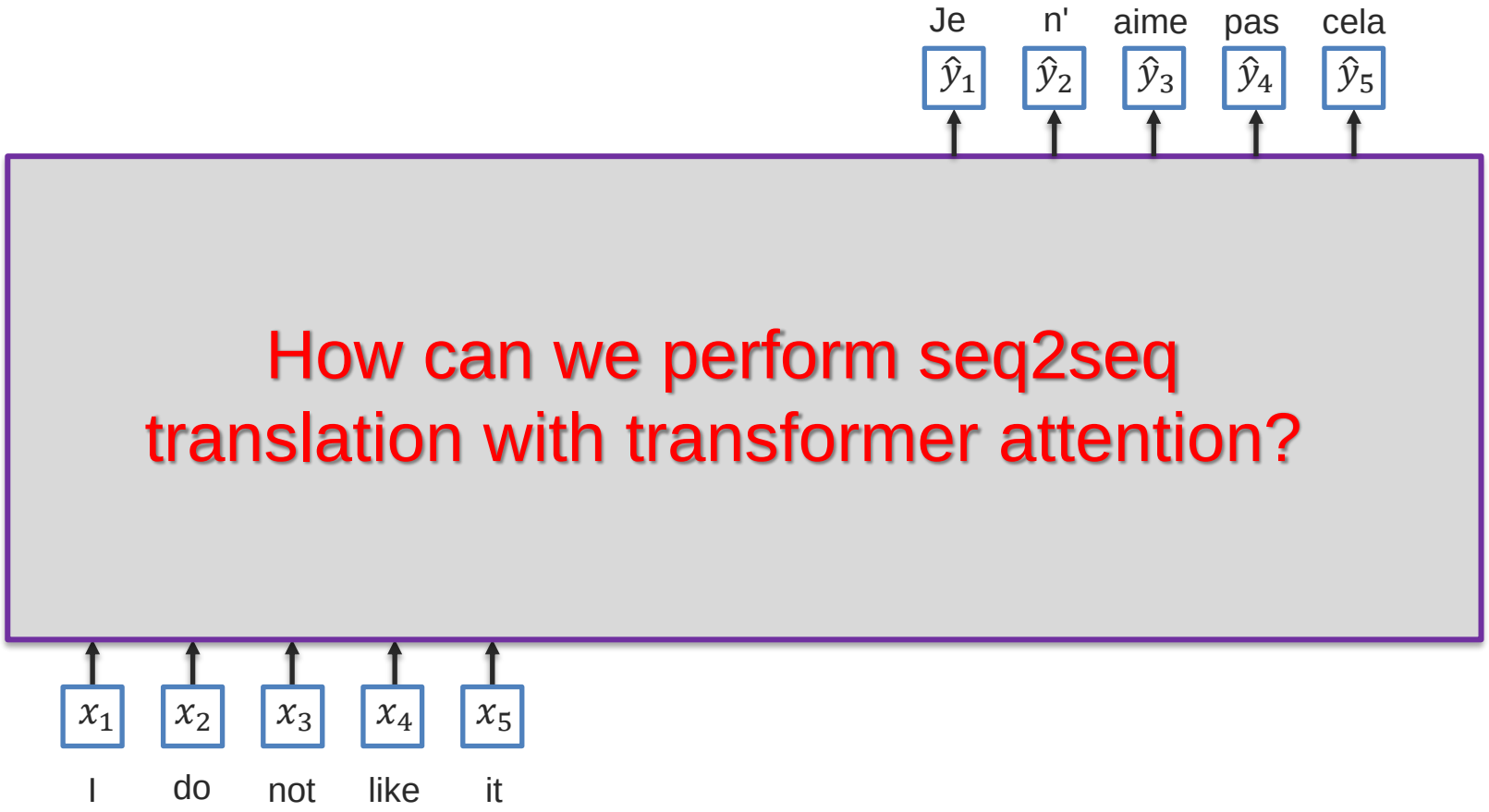


Transformer Multi-Head Attention

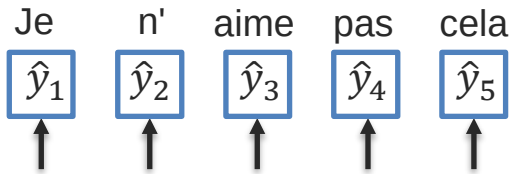


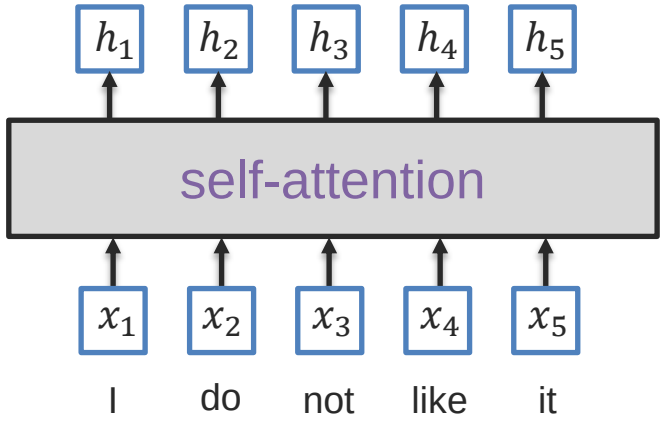
Sequence-to-Sequence Using Transformer

Sequence-to-Sequence Modeling

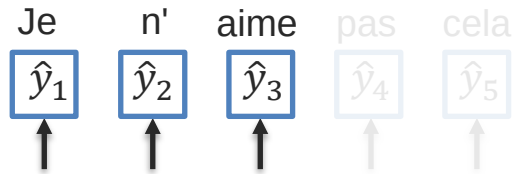


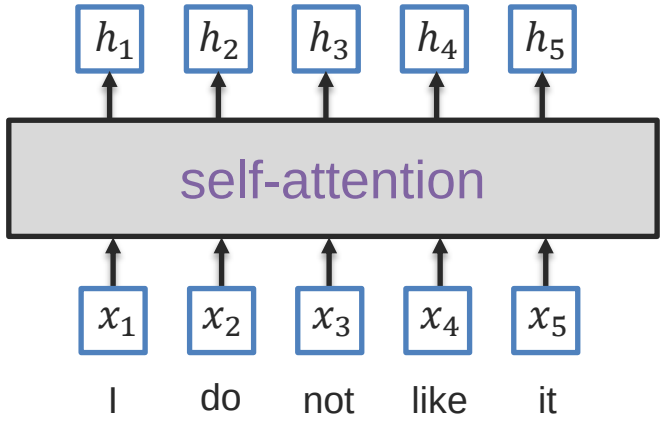
Seq2Seq with Transformer Attentions

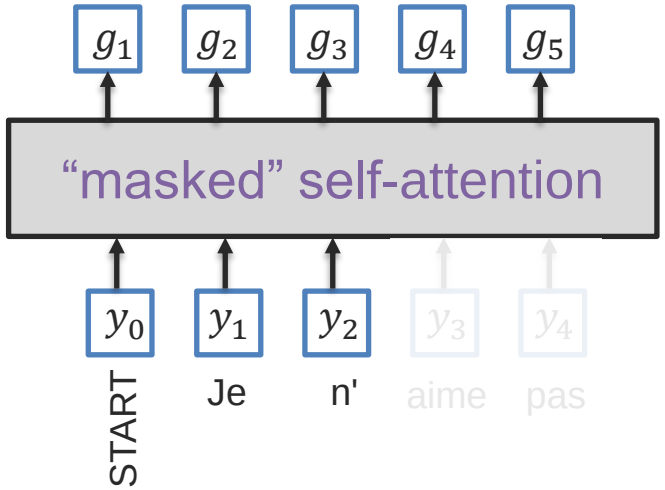




Seq2Seq with Transformer Attentions

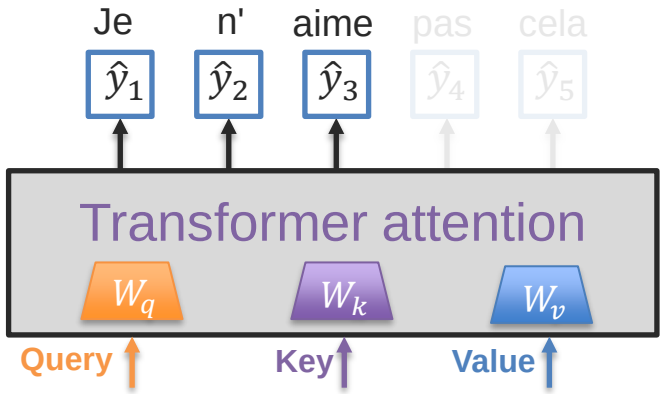


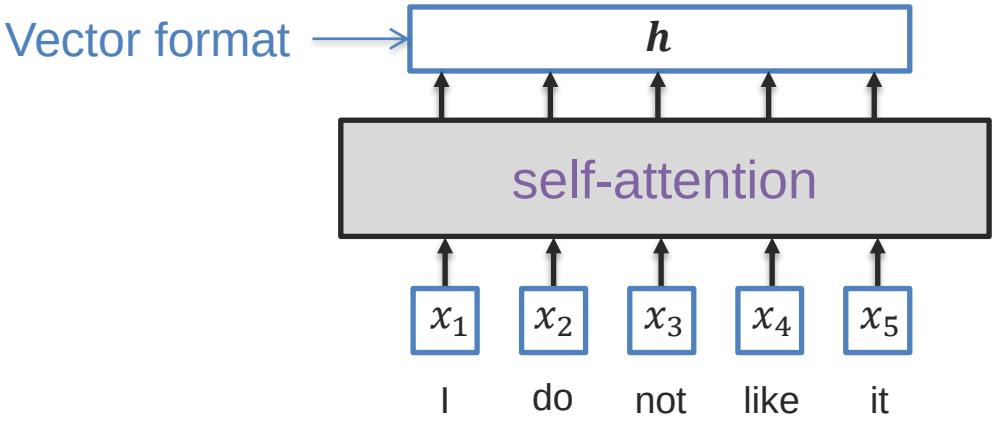


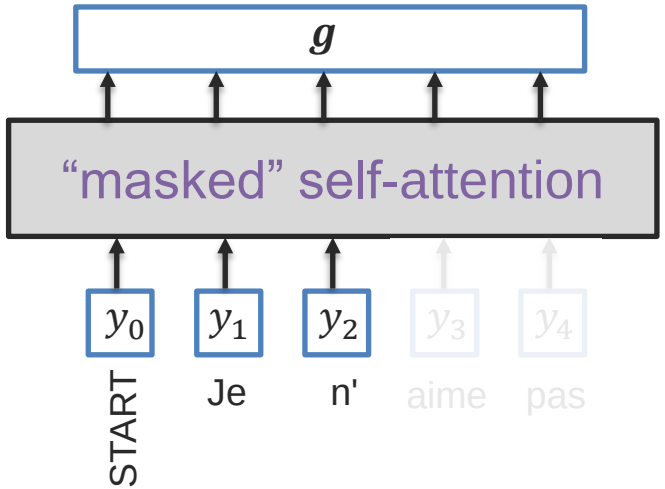


Seq2Seq with Transformer Attentions

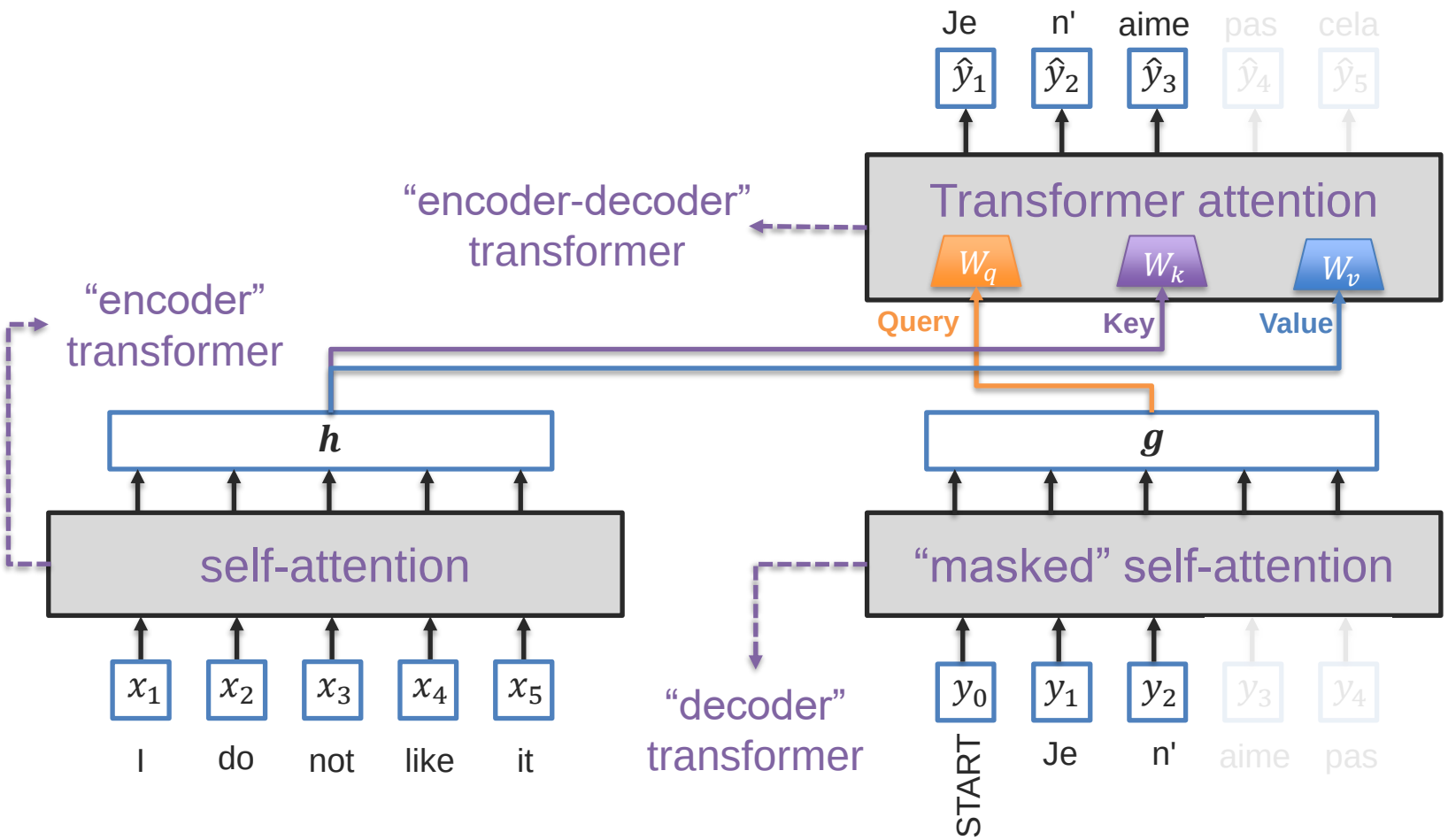
How should we connect the encoder and decoder self-attention to the transformer attention?







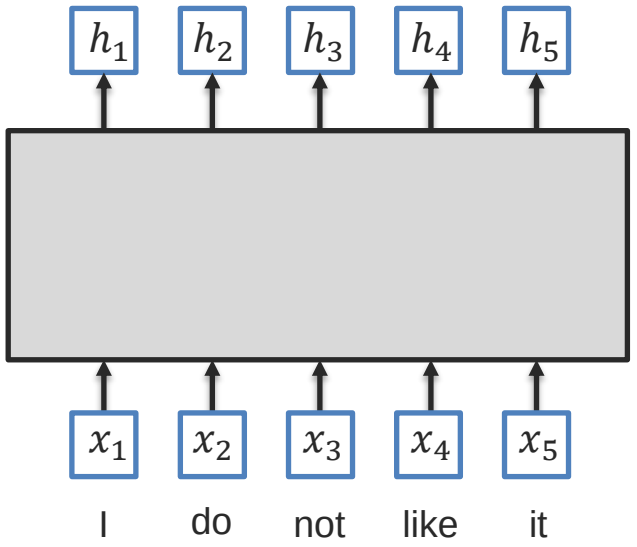
Seq2Seq with Transformer Attentions



Language Pre-training

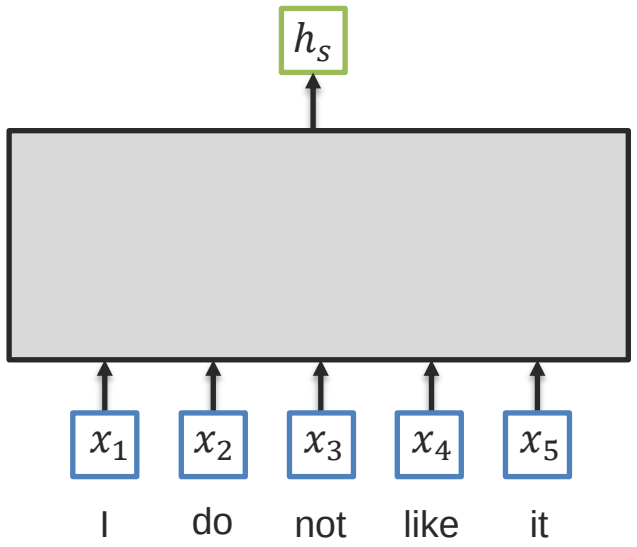
Token-level and Sentence-level Embeddings

Token-level embeddings



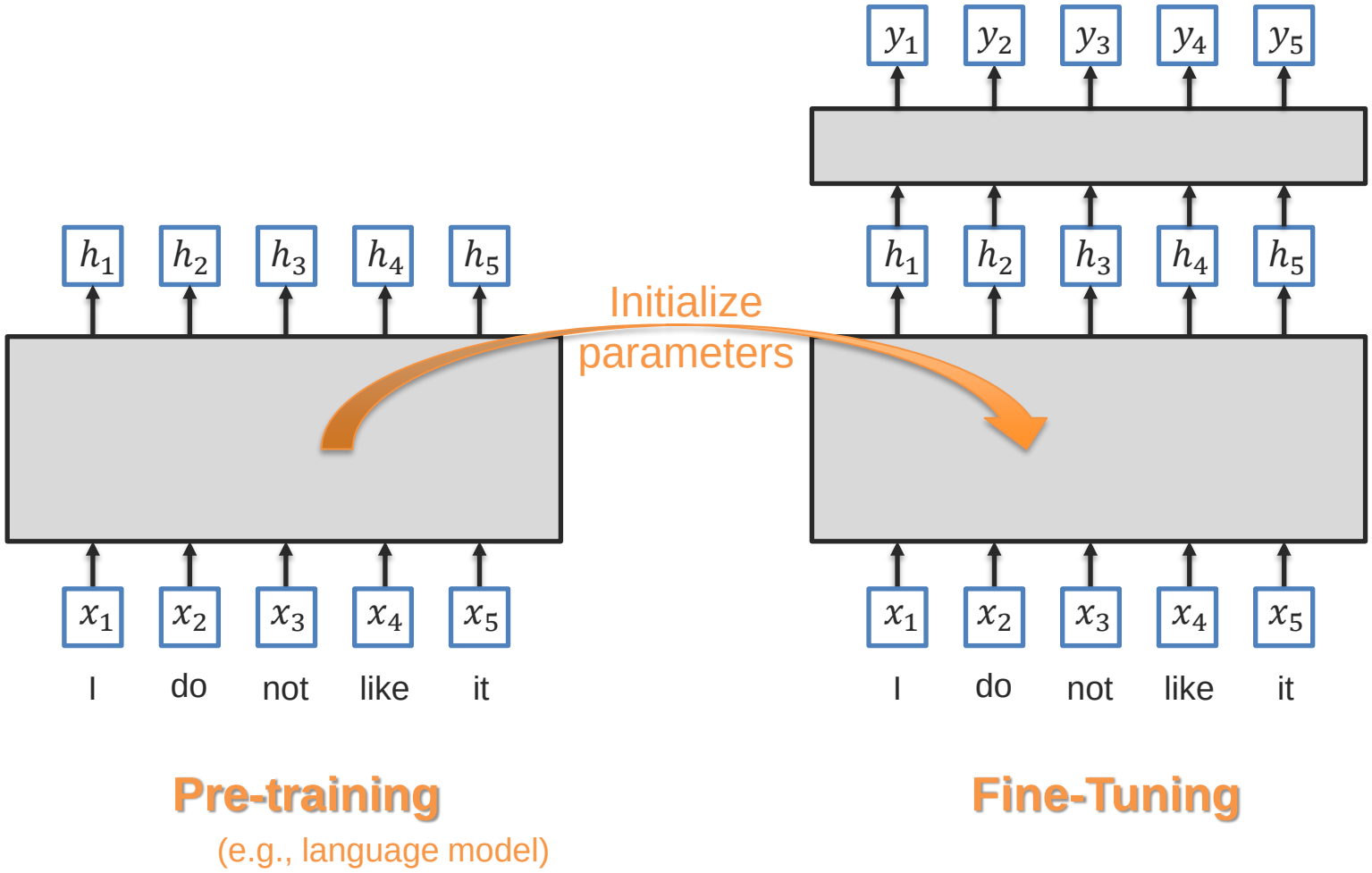
Which tasks?

Sentence-level embedding



Which tasks?

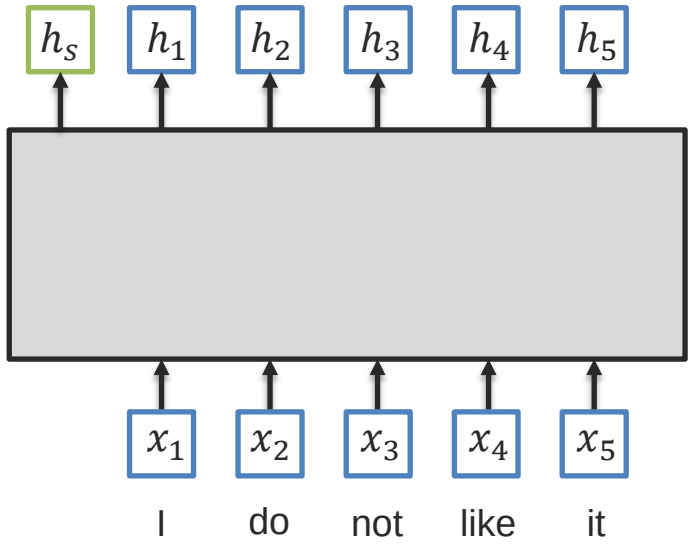
Pre-Training and Fine-Tuning



BERT: Bidirectional Encoder Representations from Transformers

Advantages:

1. Jointly learn representation for token-level and sentence level
2. Same network architecture for pre-training and fine-tuning

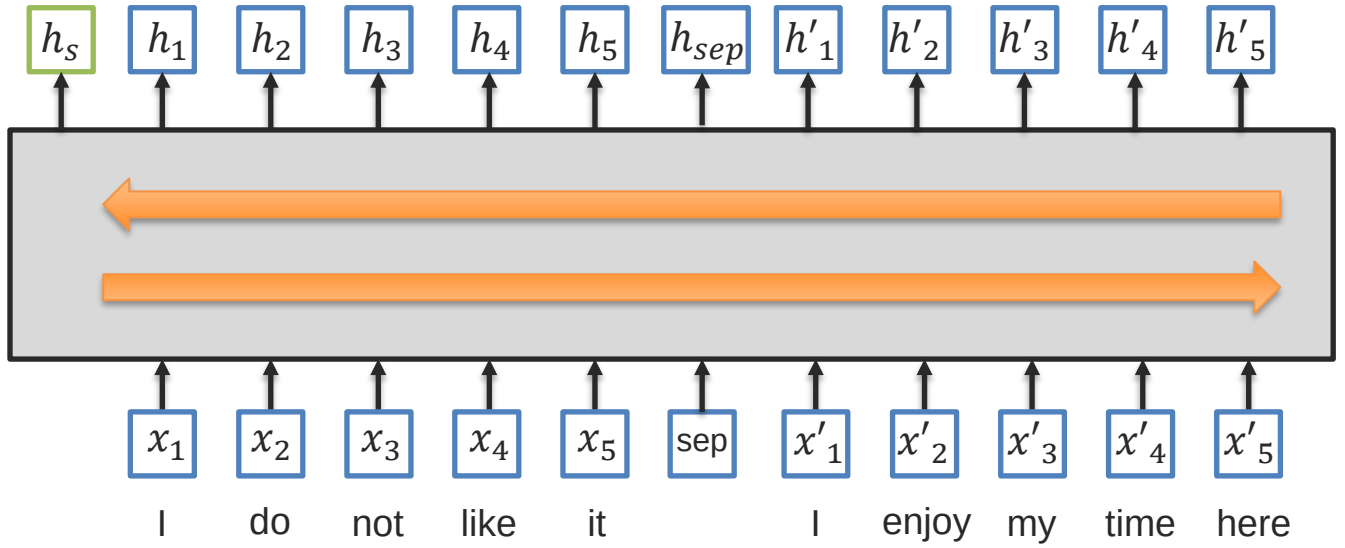


BERT: Bidirectional Encoder Representations from Transformers

Advantages:

1. Jointly learn representation for token-level and sentence level
2. Same network architecture for pre-training and fine-tuning
3. Can be used learn relationship between sentences
4. Models bidirectional and long-range interactions between tokens

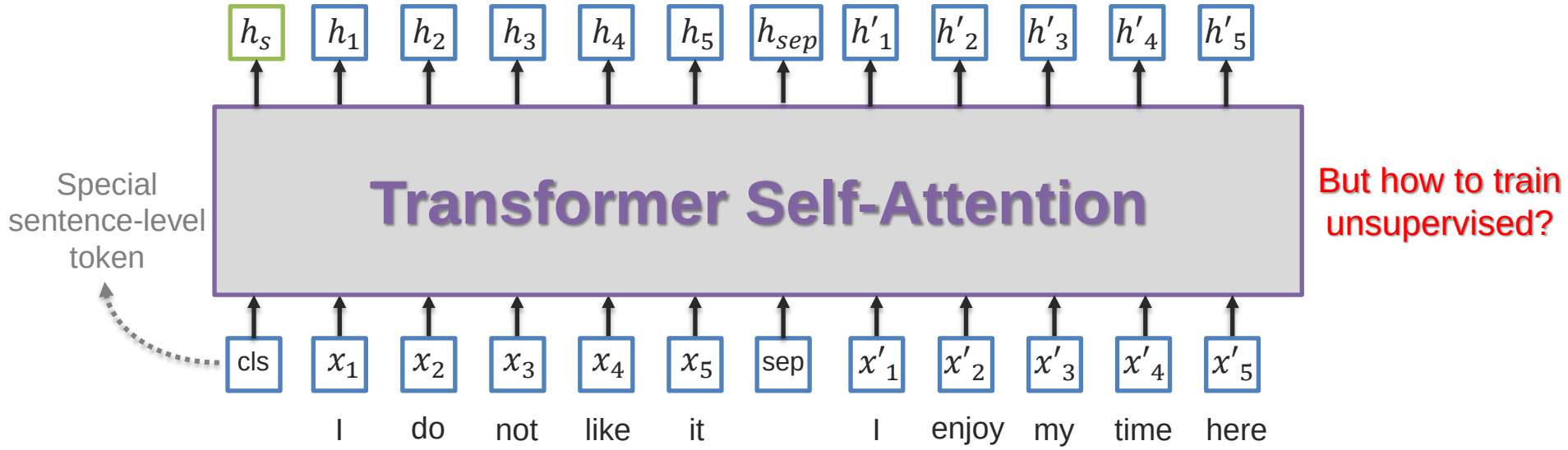
How can we do all this?



BERT: Bidirectional Encoder Representations from Transformers

Advantages:

1. Jointly learn representation for token-level and sentence level
2. Same network architecture for pre-training and fine-tuning
3. Can be used learn relationship between sentences
4. Models bidirectional interactions between tokens

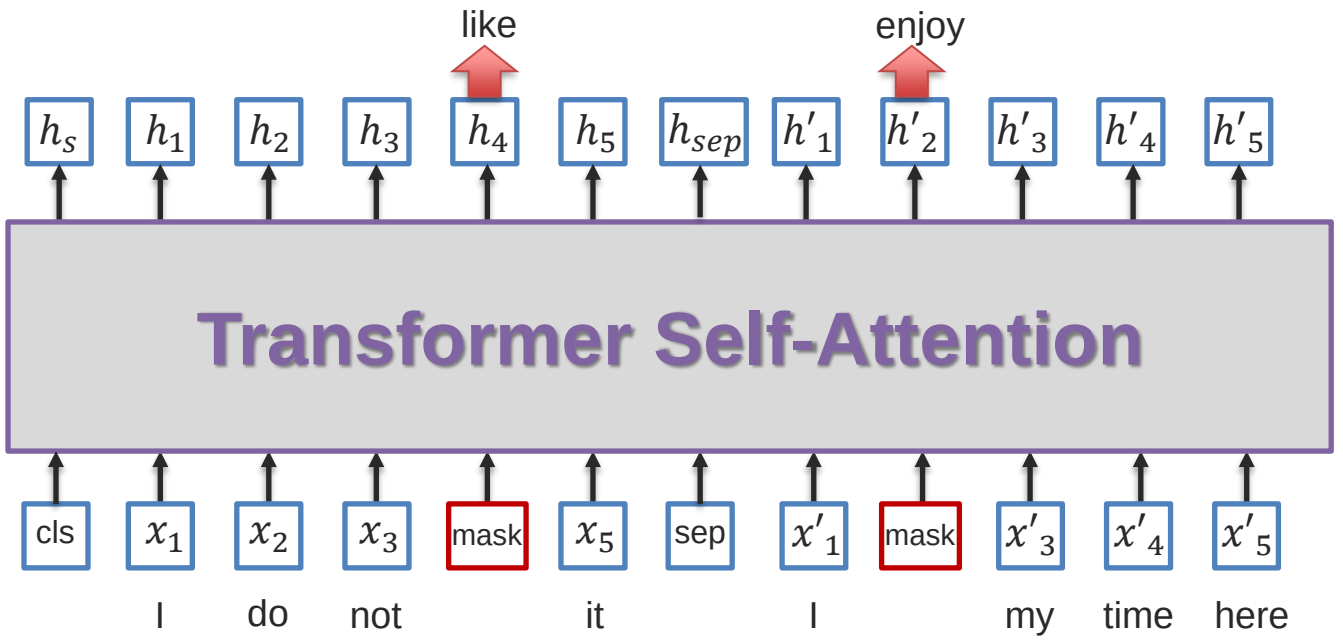


Pre-training BERT Model

1 **Masked Language Model**

Randomly mask input tokens and then try to predict them

What is the loss function?



Pre-training BERT Model

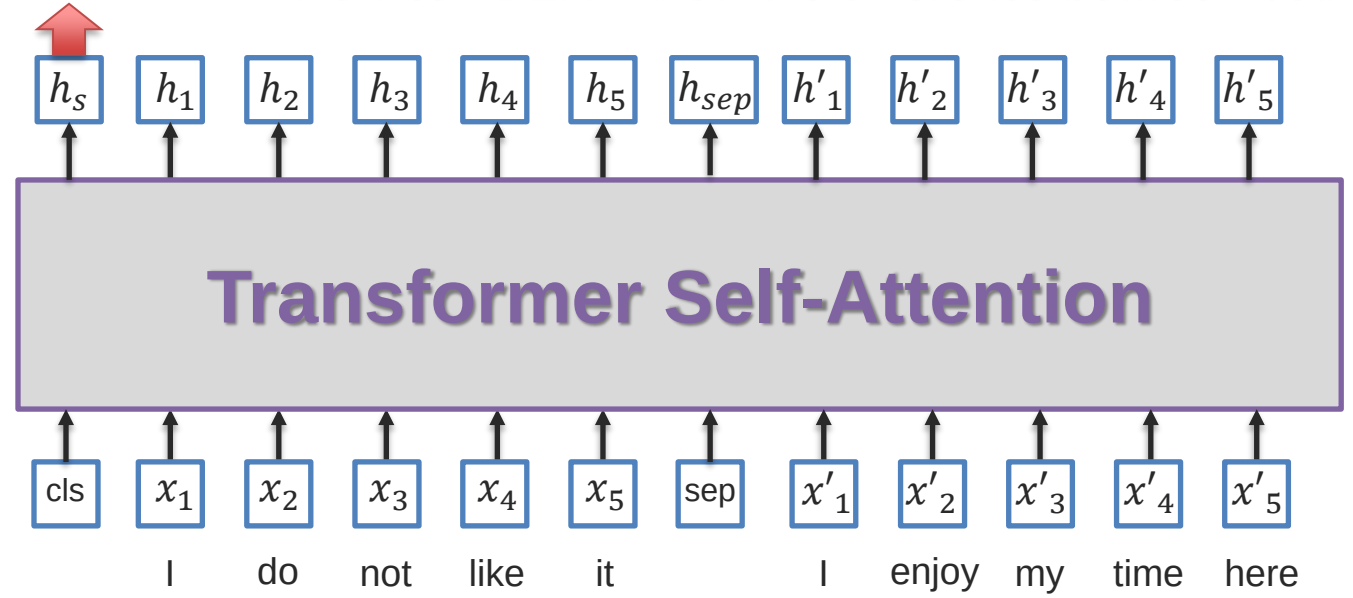
2 **Next Sentence Prediction**

Given two sentences, predict if this is the next one or not

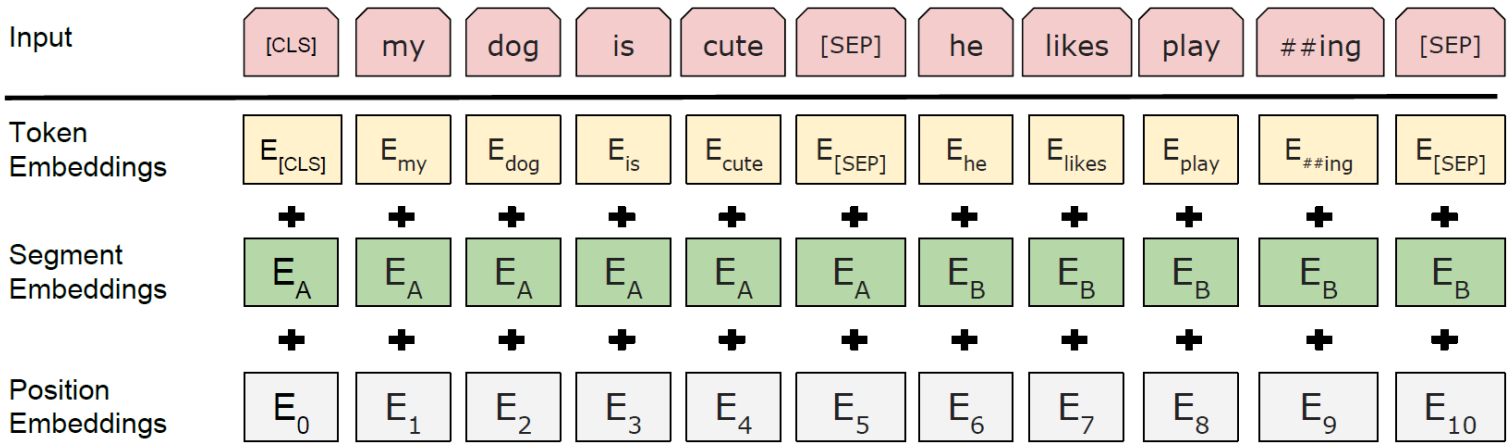
What is the loss function?

Where can we find training data?

How can BERT know the difference between both sentences?



Three Embeddings: Token + Position + Sentence

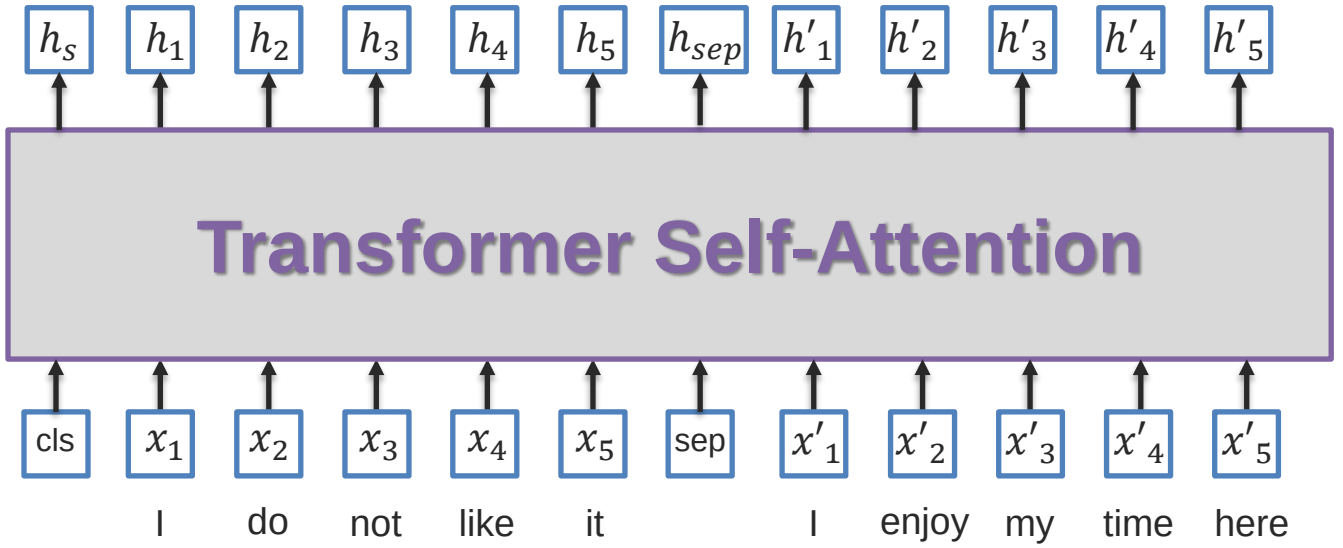


Fine-Tuning BERT

1. Sentence-level classification for only one sentence

Examples: sentiment analysis, document classification

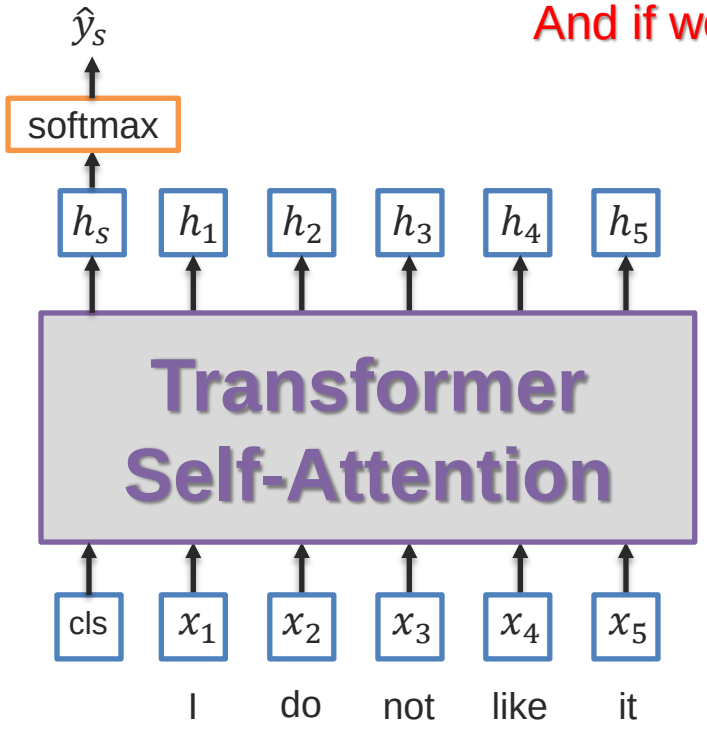
How?



Fine-Tuning BERT

1. Sentence-level classification for only one sentence

Examples: sentiment analysis, document classification

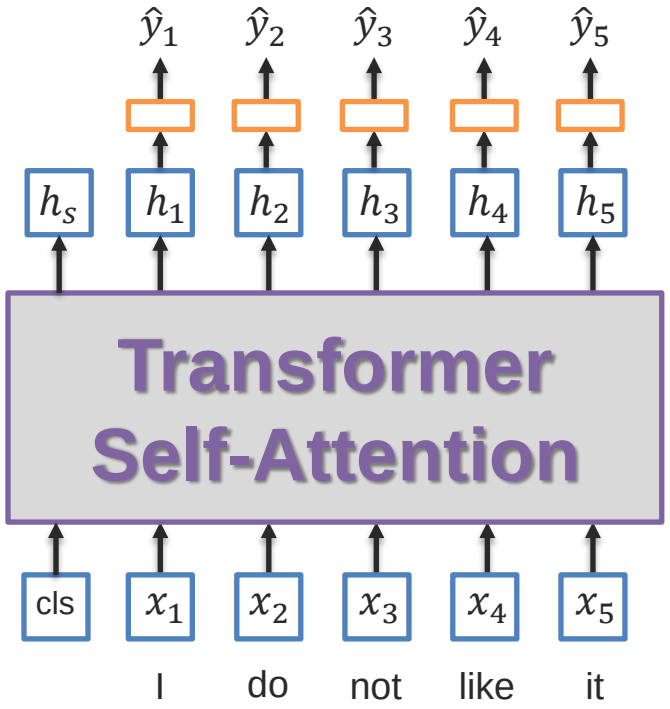


And if we have a label for each token?

Fine-Tuning BERT

2 Token-level classification for only one sentence

Examples: part-of-speech tagging, slot filling

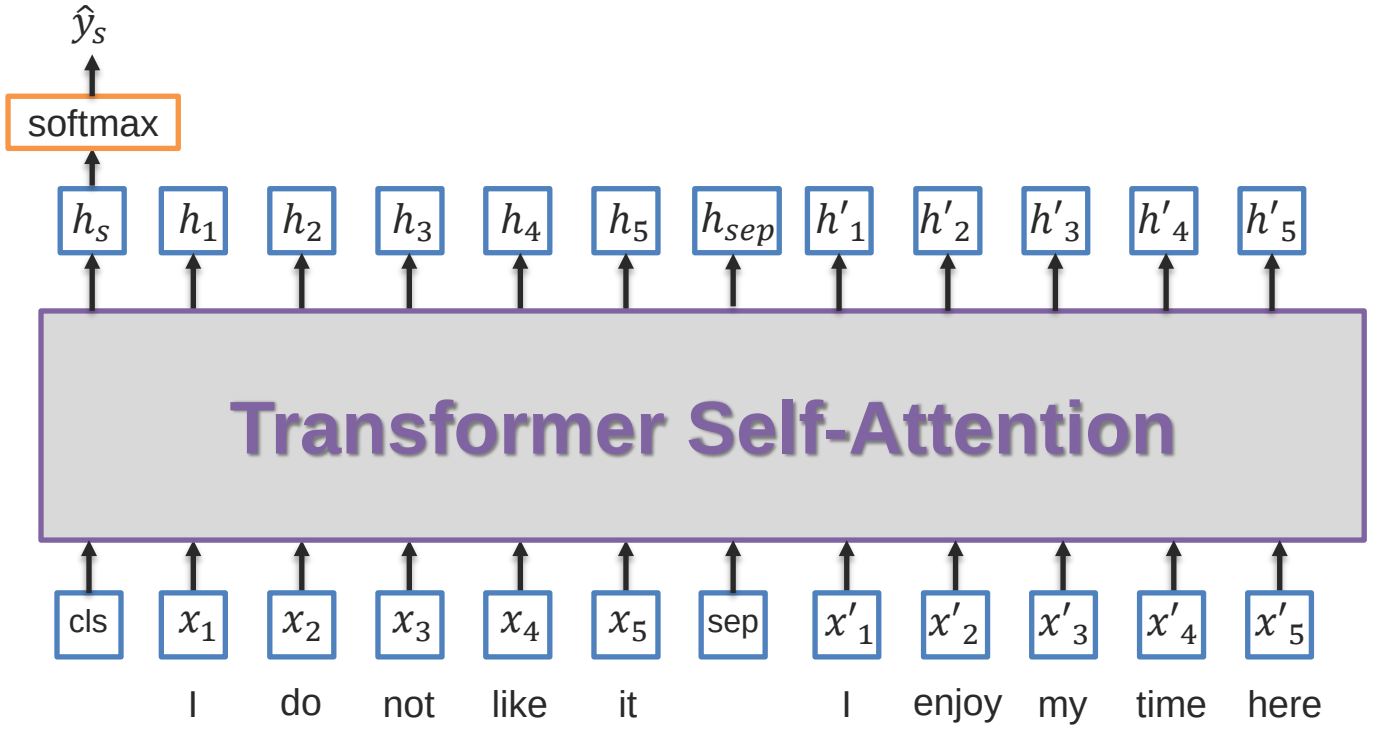


How to compare two sentences?

Fine-Tuning BERT

3 Sentence-level classification for two sentences

Examples: natural language inference



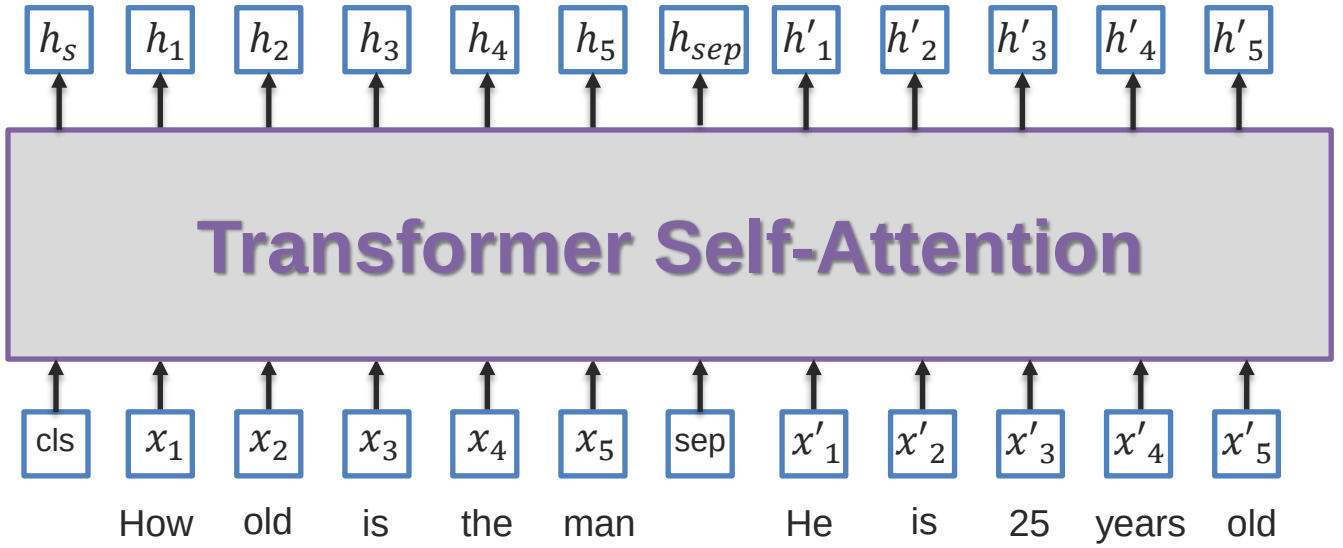
Fine-Tuning BERT

4 Question-answering: find start/end of the answer in the document

Paragraph: " . . . *Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society—the species were relatively rare—and little opposition was raised.*"

Question 1: "*Which laws faced significant opposition?*"
Plausible Answer: *later laws*

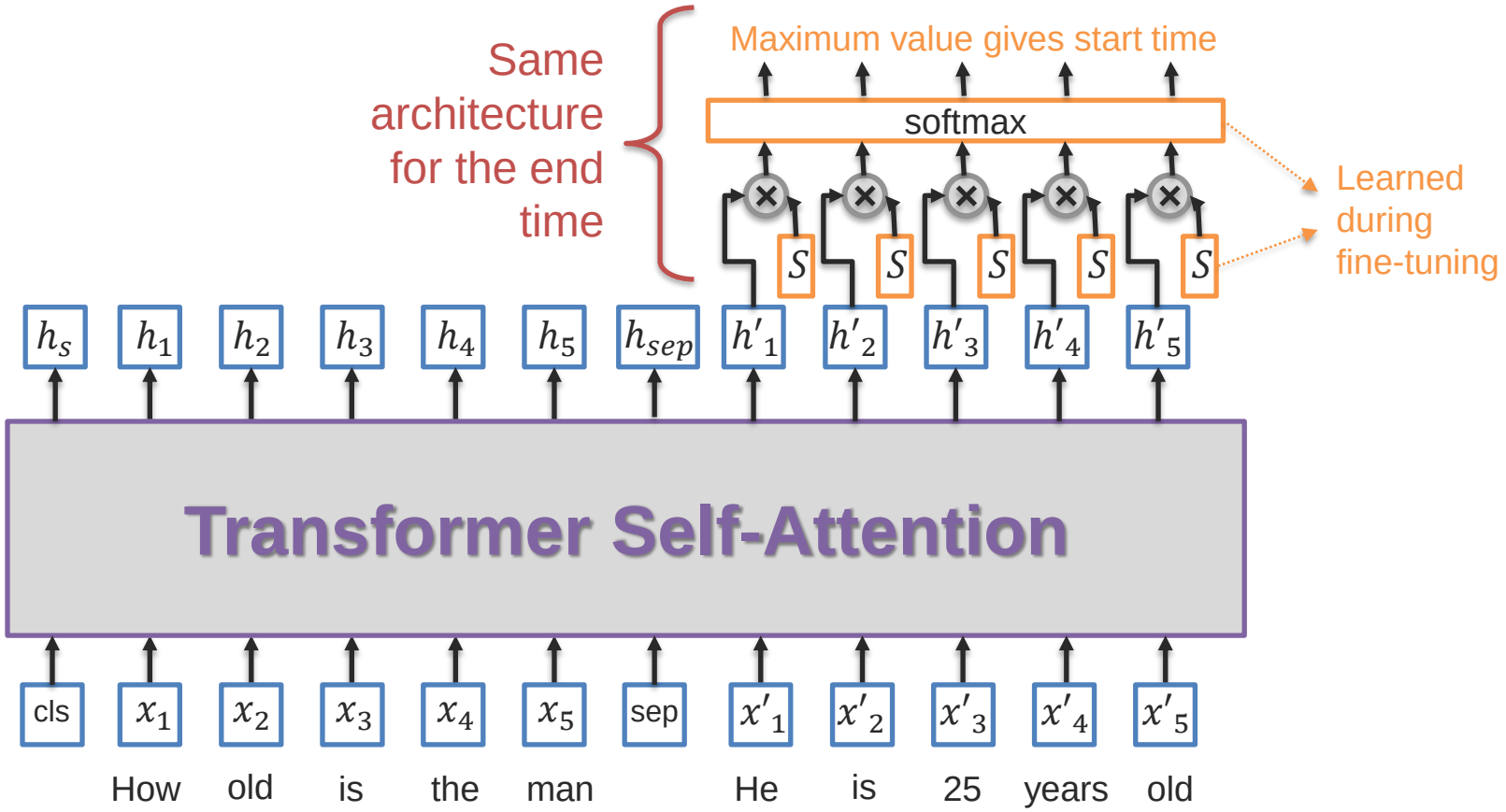
Question 2: "*What was the name of the 1937 treaty?*"
Plausible Answer: *Bald Eagle Protection Act*



How?

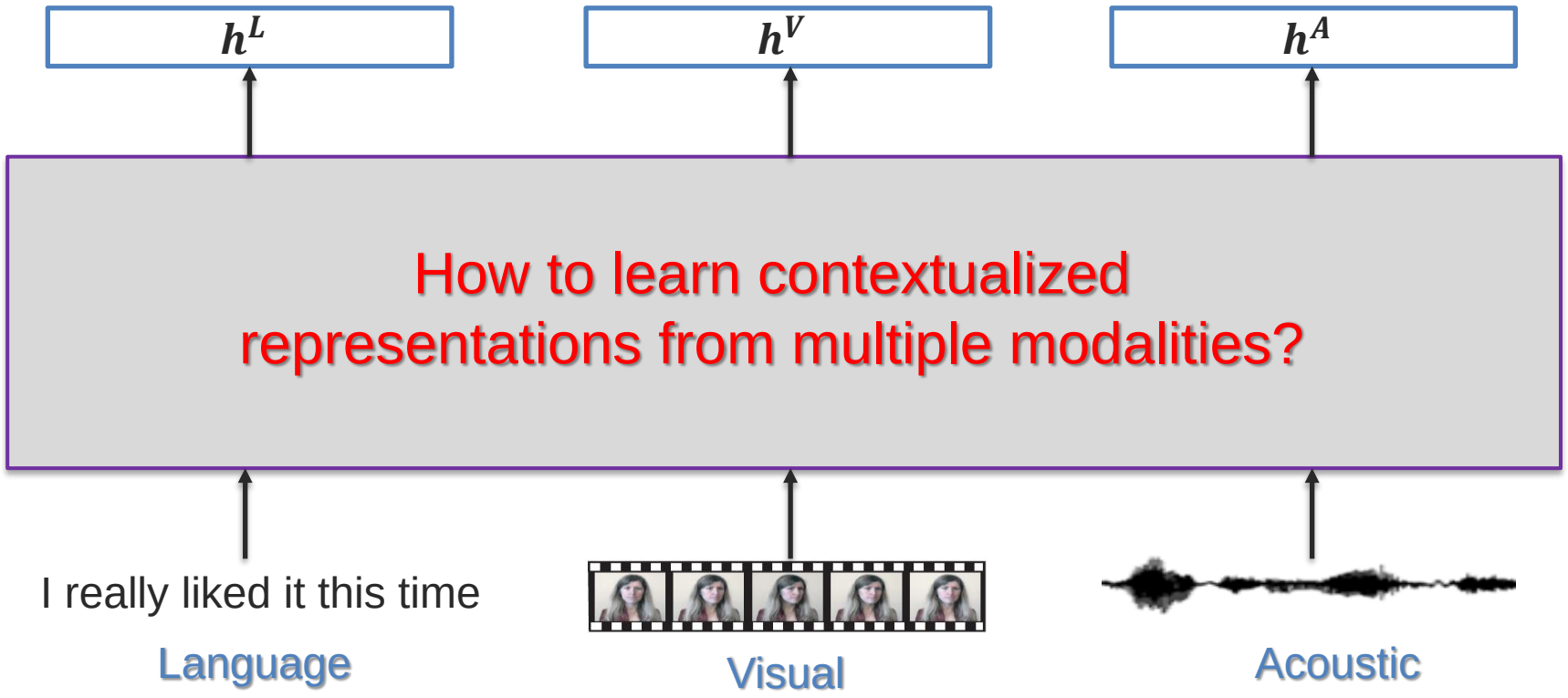
Fine-Tuning BERT

4 Question-answering: find start/end of the answer in the document

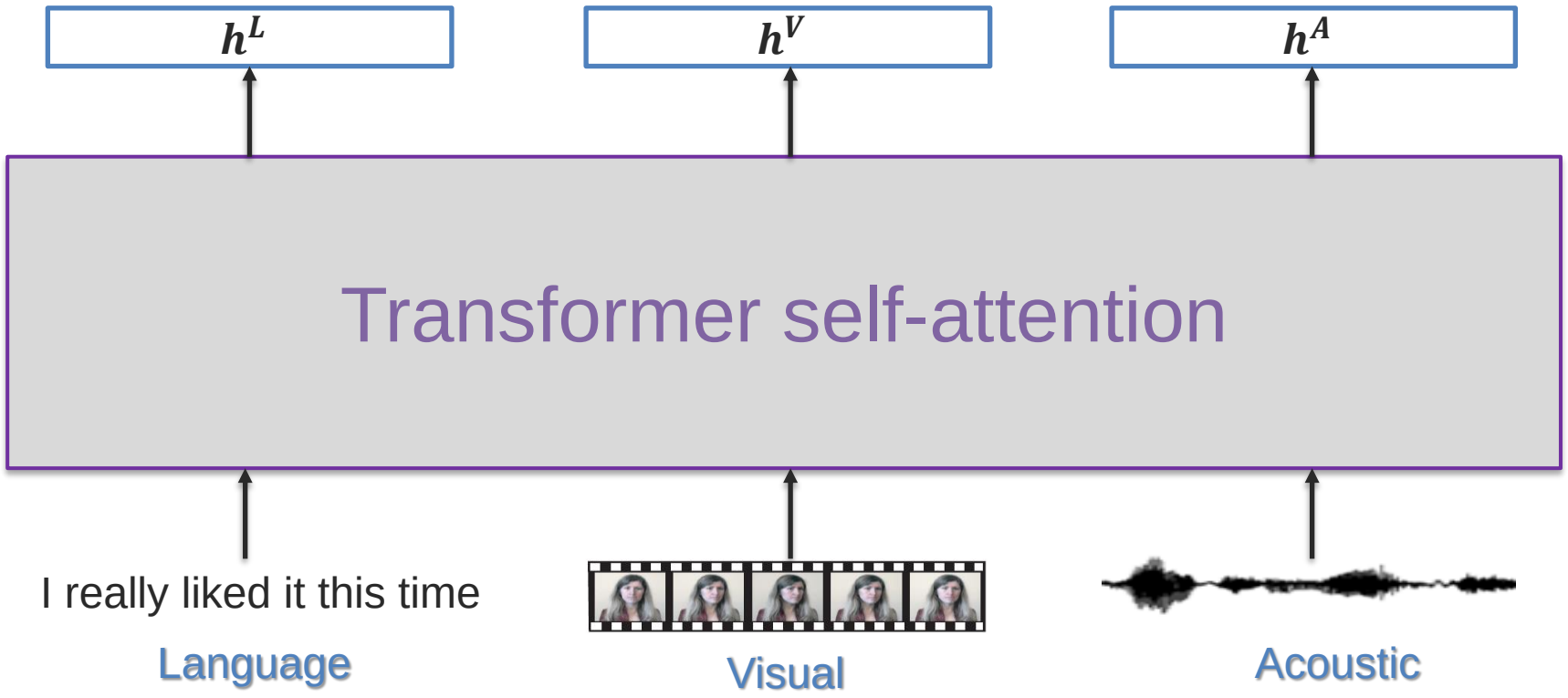


Contextualized Multimodal Embedding

Multimodal Embeddings

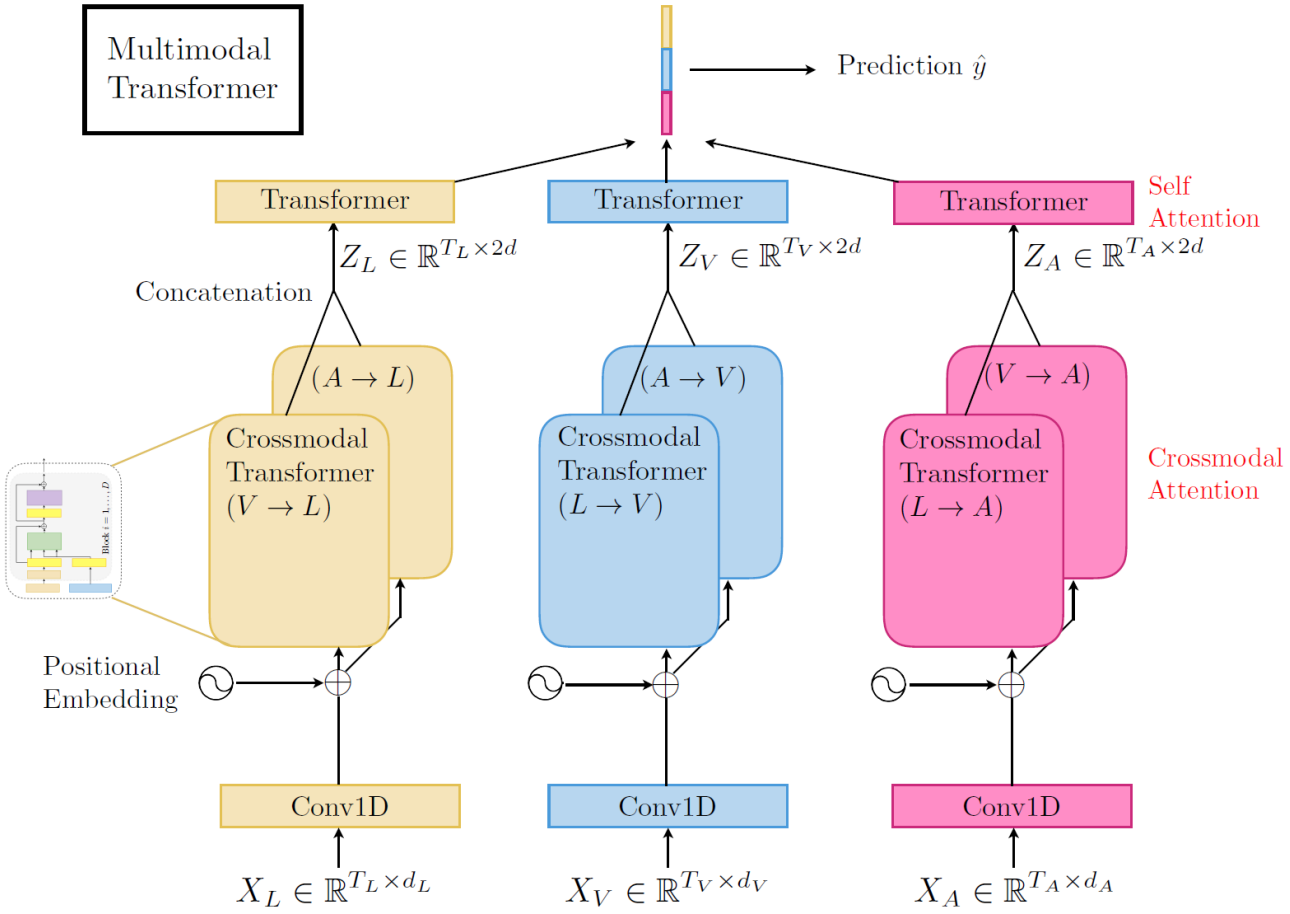


Simple Solution: Contextualized Multimodal Embeddings



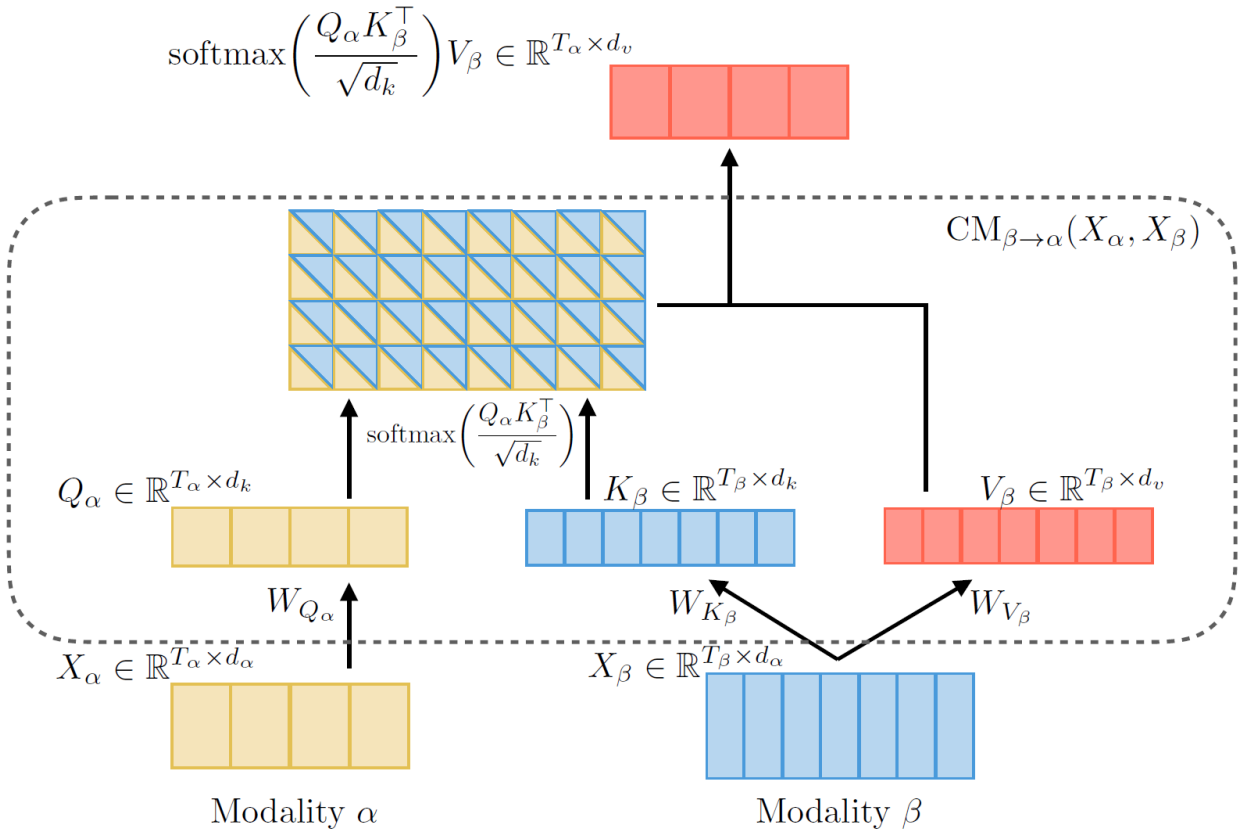
Any other approach?

Multimodal Transformer – Pairwise Cross-Modal



Tsai et al., Multimodal Transformer for Unaligned Multimodal Language Sequences, ACL 2019

Cross-Modal Transformer



Tsai et al., Multimodal Transformer for Unaligned Multimodal Language Sequences, ACL 2019

And Many More… Next week!