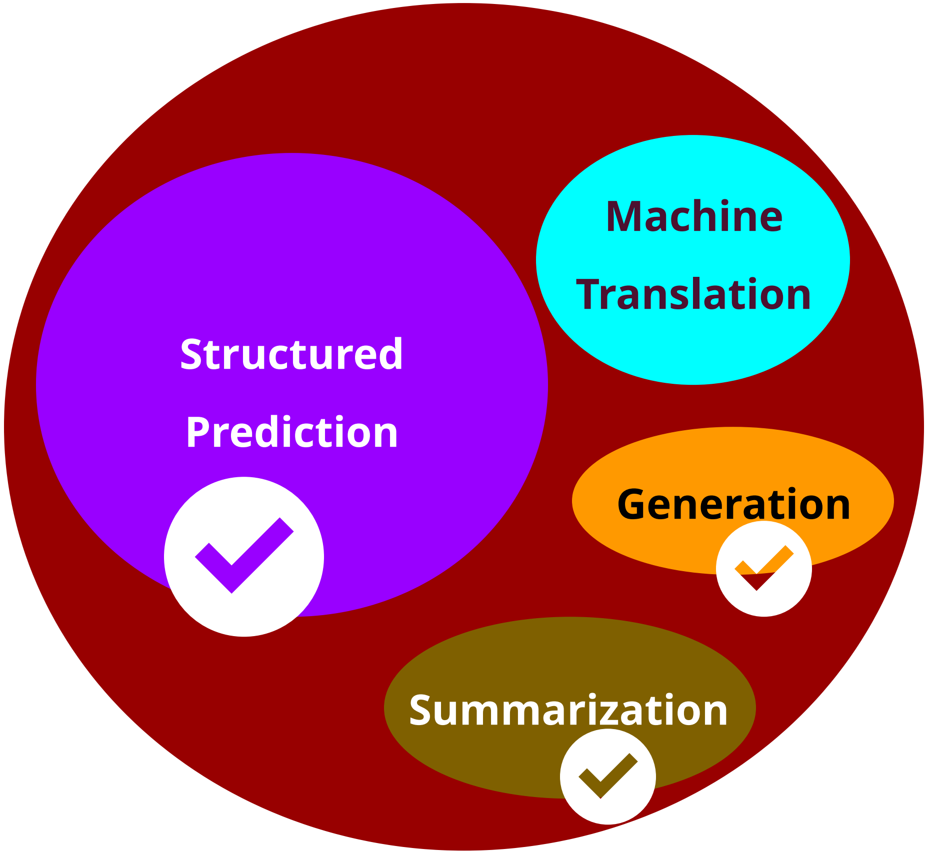
50.040 Natural Language Processing

Lu, Wei

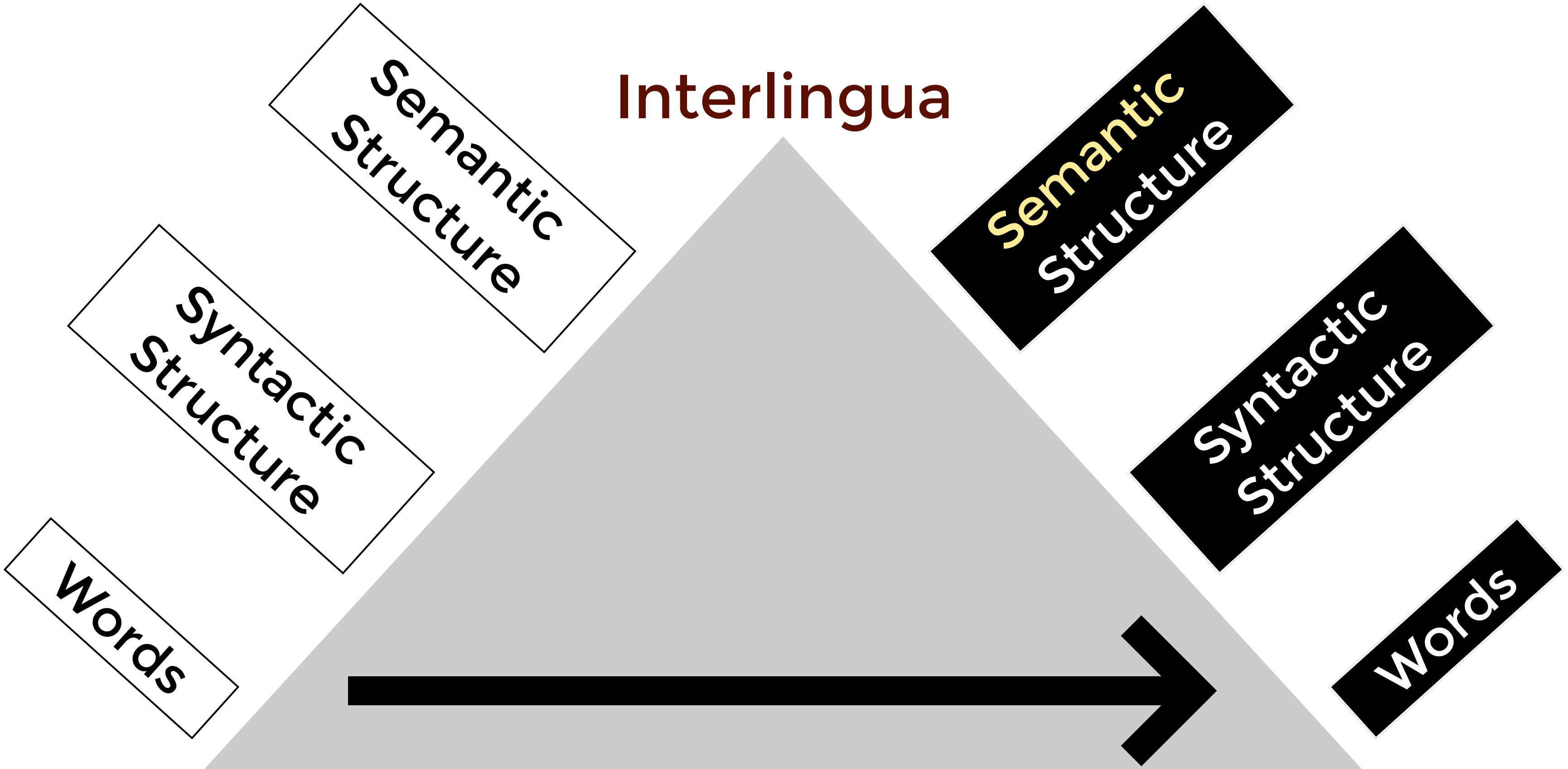
SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

Tasks in NLP



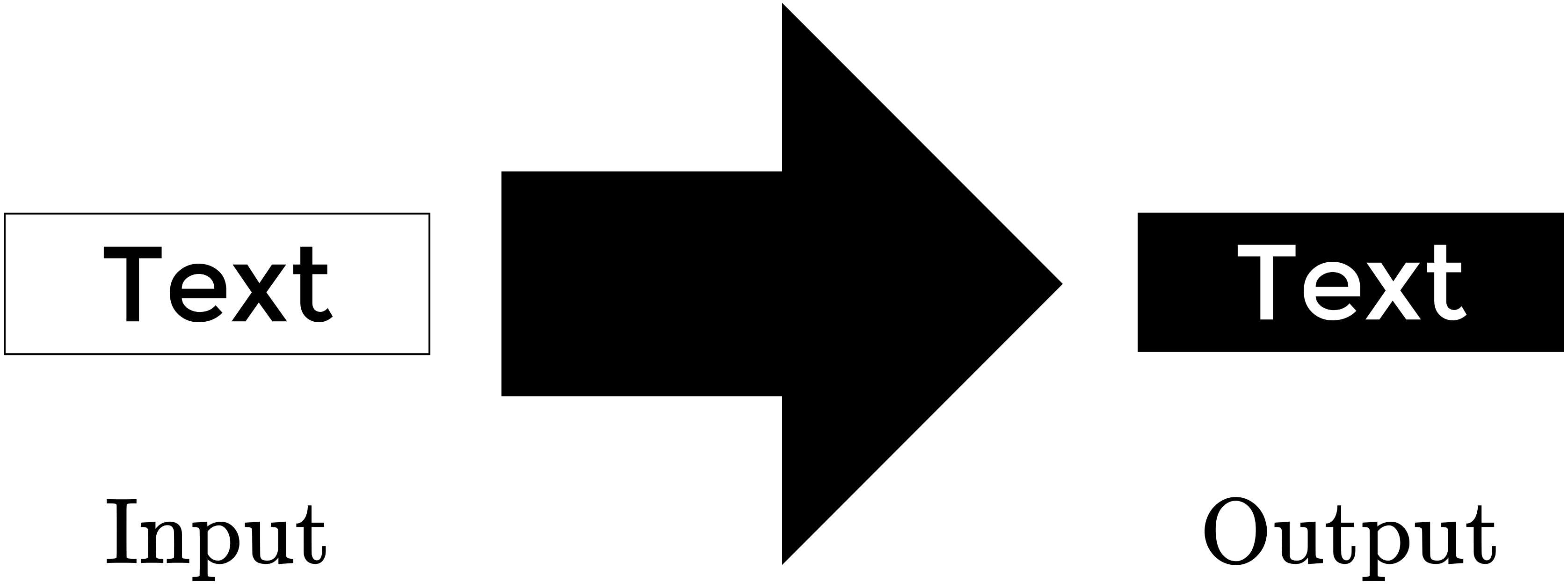
Supervised

Machine Translation

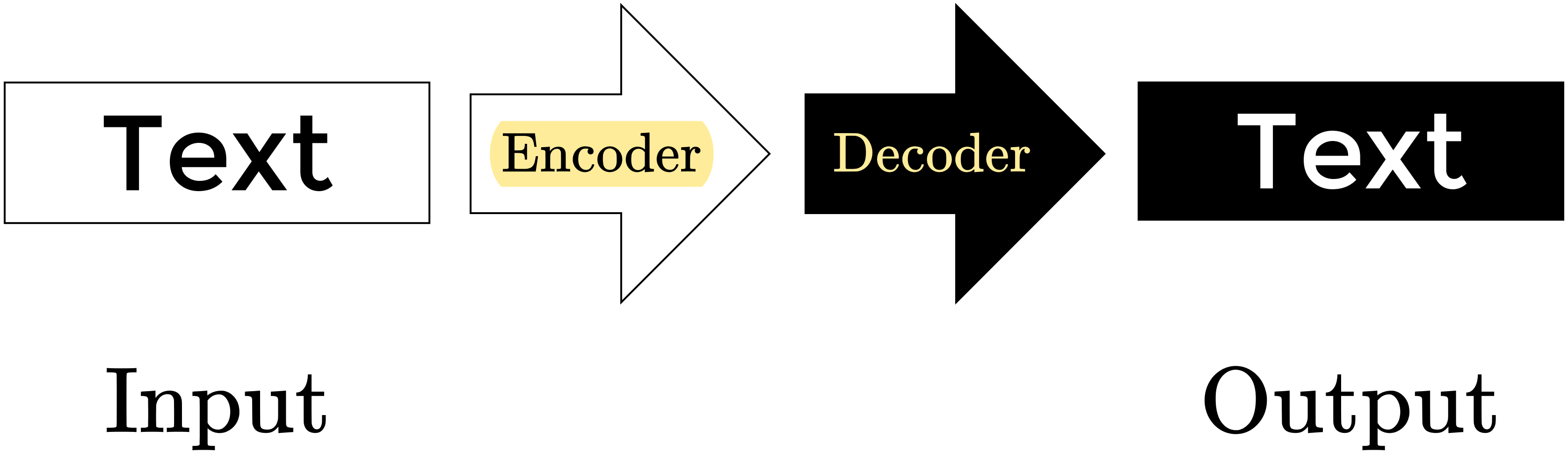


Text-to-text Problem

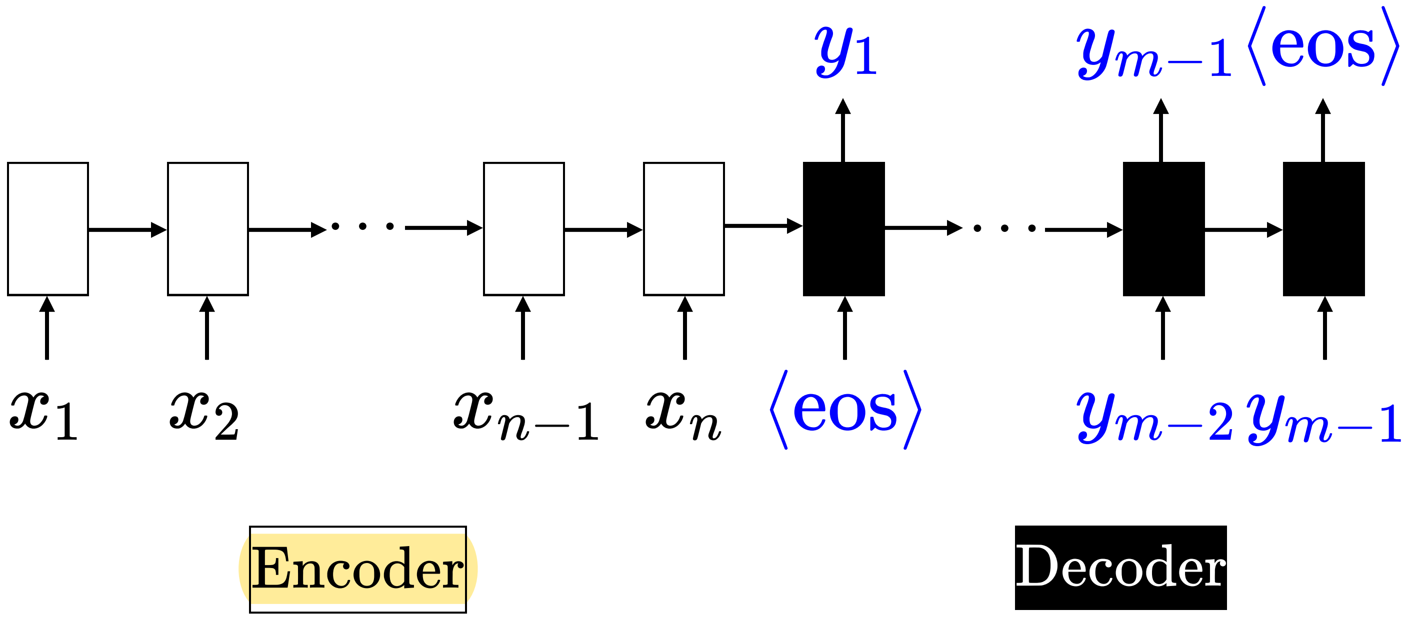
Text-to-Text Generation



Encoder-Decoder

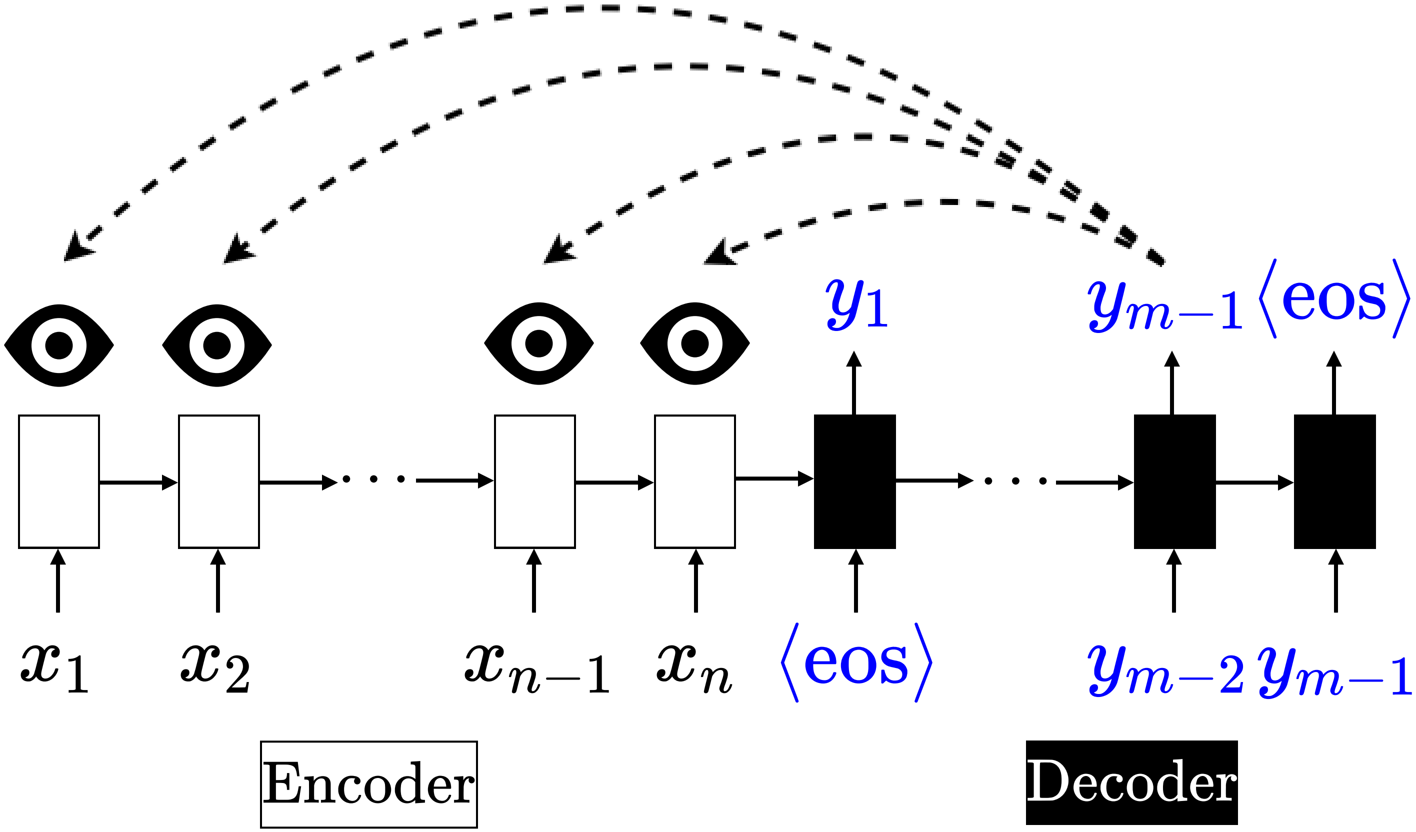


Sequence to Sequence



$$p(y_1, \ldots, y_m | x_1, \ldots, x_n)$$

Seq2Seq with Attention
(Bahdanau et al. 2015)



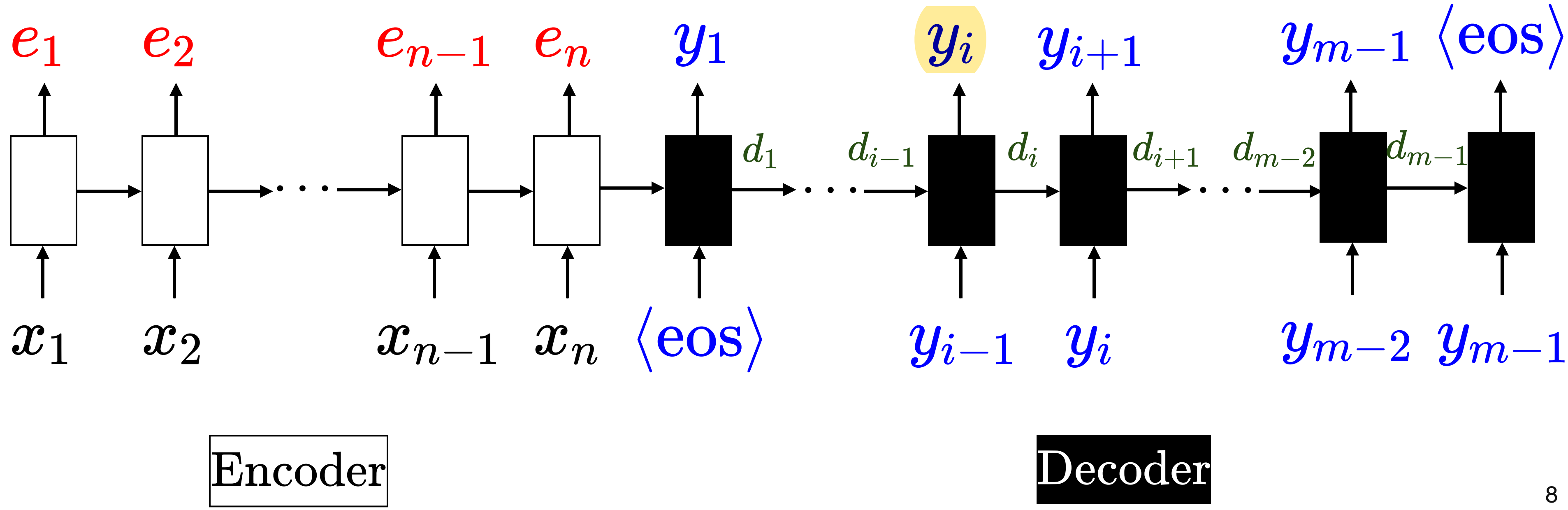
Seq2Seq with Attention

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_i)$$

$$a_j^i = \text{softmax}(u_j^i)$$

$$d_i' = \sum_{j=1}^{n} a_j^i e_j$$

A weighted combination of $e_1 \ldots e_n$



Attention

Cosine

$$\frac{{d_i}^T e_j}{||d_i||\cdot||e_j||}$$

(Graves et al. 2014)

Concatenation

$$v^T \tanh(W[e_j; d_i])$$

(Bahdanau et al. 2015)

Roughly speaking, they are used for measuring the degree of similarity or correlation between encoder and decoder representation states

Dot-Product

$${d_i}^T e_j$$

(Luong et al. 2015)

General Product

$${d_i}^T W e_j$$

(Luong et al. 2015)

Attention

Cosine

$$\frac{{d_i}^T e_j}{||d_i|| \cdot ||e_j||}$$

(Graves et al. 2014)

Concatenation

$$v^T \tanh(W[e_j; d_i])$$

(Bahdanau et al. 2015)

Scaled Product

$$\frac{{d_i}^T e_j}{\sqrt{\delta}}$$

(Vaswani et al. 2017)

Dot-Product

The δ is the dimension of d_i or e_j

(Luong et al. 2015)

General Product

(Luong et al. 2015)

Attention

$$d_i^T \qquad \begin{bmatrix} - & e_1^T & - \\ - & e_2^T & - \\ \vdots & \vdots & \vdots \\ - & e_n^T & - \end{bmatrix}$$

We are interested in the similarity between d_i and e_j

Attention

Query

$$d_i^T$$

Key

$$\begin{bmatrix} - & e_1^T & - \\ - & e_2^T & - \\ \vdots & \vdots & \vdots \\ - & e_n^T & - \end{bmatrix}$$

We are interested in the similarity between our "query" and each "key"

Attention

Query

$$d_i^T$$

Key

$$\begin{bmatrix} - & e_1^T & - \\ - & e_2^T & - \\ \vdots & \vdots & \vdots \\ - & e_n^T & - \end{bmatrix}$$

The keys can be used to open the cabinet which stores "values"

Attention

Query	Key	Value
d_i^T	$\begin{bmatrix} - & e_1^T & - \\ - & e_2^T & - \\ \vdots & \vdots & \vdots \\ - & e_n^T & - \end{bmatrix}$	$\begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ \vdots & \vdots & \vdots \\ - & v_n^T & - \end{bmatrix}$
	K	V

We will obtain a weighted combination of $v_1 \ldots v_n$

Scaled Dot Product

$$\begin{bmatrix} 0.01 & 0.21 & 0.15 & \dots & 0.65 \end{bmatrix} \times$$

$$\Uparrow$$

softmax

$$\begin{bmatrix} -0.21 & 0.14 & 0.09 & \dots & 1.25 \end{bmatrix}$$

$$\Uparrow$$

$$\times \frac{1}{\sqrt{\delta}}$$

$$\begin{bmatrix} -2.1 & 1.4 & 0.9 & \dots & 12.5 \end{bmatrix}$$

$$\Uparrow$$

$$d_i^T \times$$

$$\begin{bmatrix} | & | & | & \dots & | \\ e_1 & e_2 & e_3 & \dots & e_n \\ | & | & | & \dots & | \end{bmatrix} \quad K^T$$

$$\begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ \vdots & \vdots & \vdots \\ - & v_n^T & - \end{bmatrix} \quad V$$

$$\Downarrow$$

The content you have managed to retrive from the source.

$$\boxed{\mathrm{softmax}\left(\frac{d_i^T K^T}{\sqrt{\delta}}\right) V}$$

Attention

Query Key Value

$$
\begin{bmatrix} - & d_1^T & - \\ - & d_2^T & - \\ \vdots & \vdots & \vdots \\ - & d_m^T & - \end{bmatrix}
\quad
\begin{bmatrix} - & e_1^T & - \\ - & e_2^T & - \\ \vdots & \vdots & \vdots \\ - & e_n^T & - \end{bmatrix}
\quad
\begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ \vdots & \vdots & \vdots \\ - & v_n^T & - \end{bmatrix}
$$

Q K V

If we would like to process multiple queries simultaneously

Attention

Query	Key	Value
$\begin{bmatrix} - & d_1^T & - \\ - & d_2^T & - \\ \vdots & \vdots & \vdots \\ - & d_m^T & - \end{bmatrix}$	$\begin{bmatrix} - & e_1^T & - \\ - & e_2^T & - \\ \vdots & \vdots & \vdots \\ - & e_n^T & - \end{bmatrix}$	$\begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ \vdots & \vdots & \vdots \\ - & v_n^T & - \end{bmatrix}$
Q	K	V

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{\delta}}\right)V$$

Scaled Dot Product

$$\begin{bmatrix} 0.01 & 0.21 & 0.15 & \dots & 0.65 \end{bmatrix} \times \begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ \vdots & \vdots & \vdots \\ - & v_n^T & - \end{bmatrix} V$$

$$\Uparrow$$

$$\text{softmax}$$

$$\begin{bmatrix} -0.21 & 0.14 & 0.09 & \dots & 1.25 \end{bmatrix}$$

$$\Uparrow$$

$$\times \frac{1}{\sqrt{\delta}}$$

$$\begin{bmatrix} -2.1 & 1.4 & 0.9 & \dots & 12.5 \end{bmatrix}$$

$$\Uparrow$$

$$Q \times$$

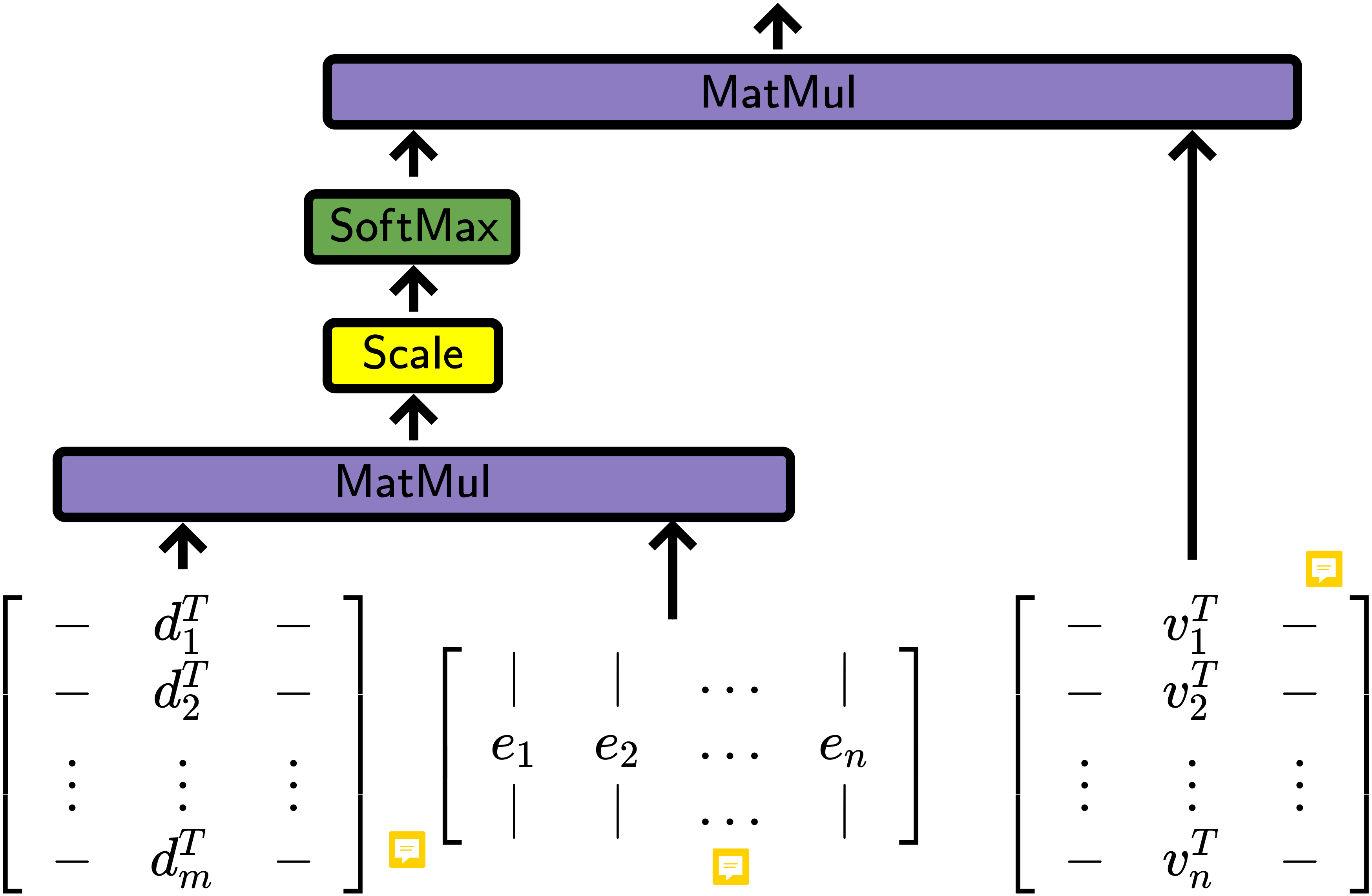
$$\begin{bmatrix} | & | & | & \dots & | \\ e_1 & e_2 & e_3 & \dots & e_n \\ | & | & | & \dots & | \end{bmatrix} K^T$$

$$\Downarrow$$

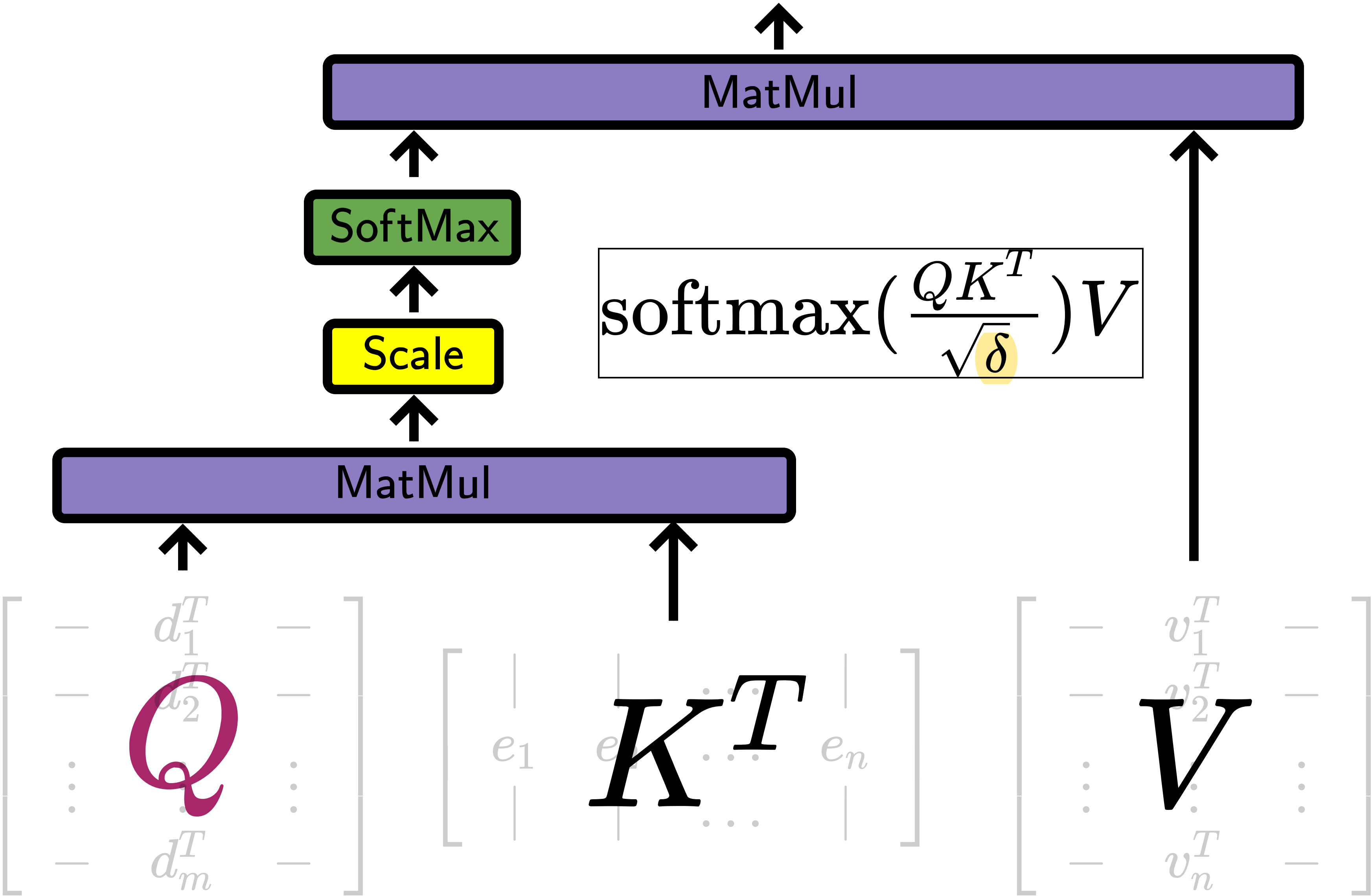
The content you have managed to retrive from the source.

$$\boxed{\text{softmax}\left(\frac{QK^T}{\sqrt{\delta}}\right)V}$$

Scaled Dot Product



Scaled Dot Product



Machine Translation

seq2seq

recurrent network based encoder

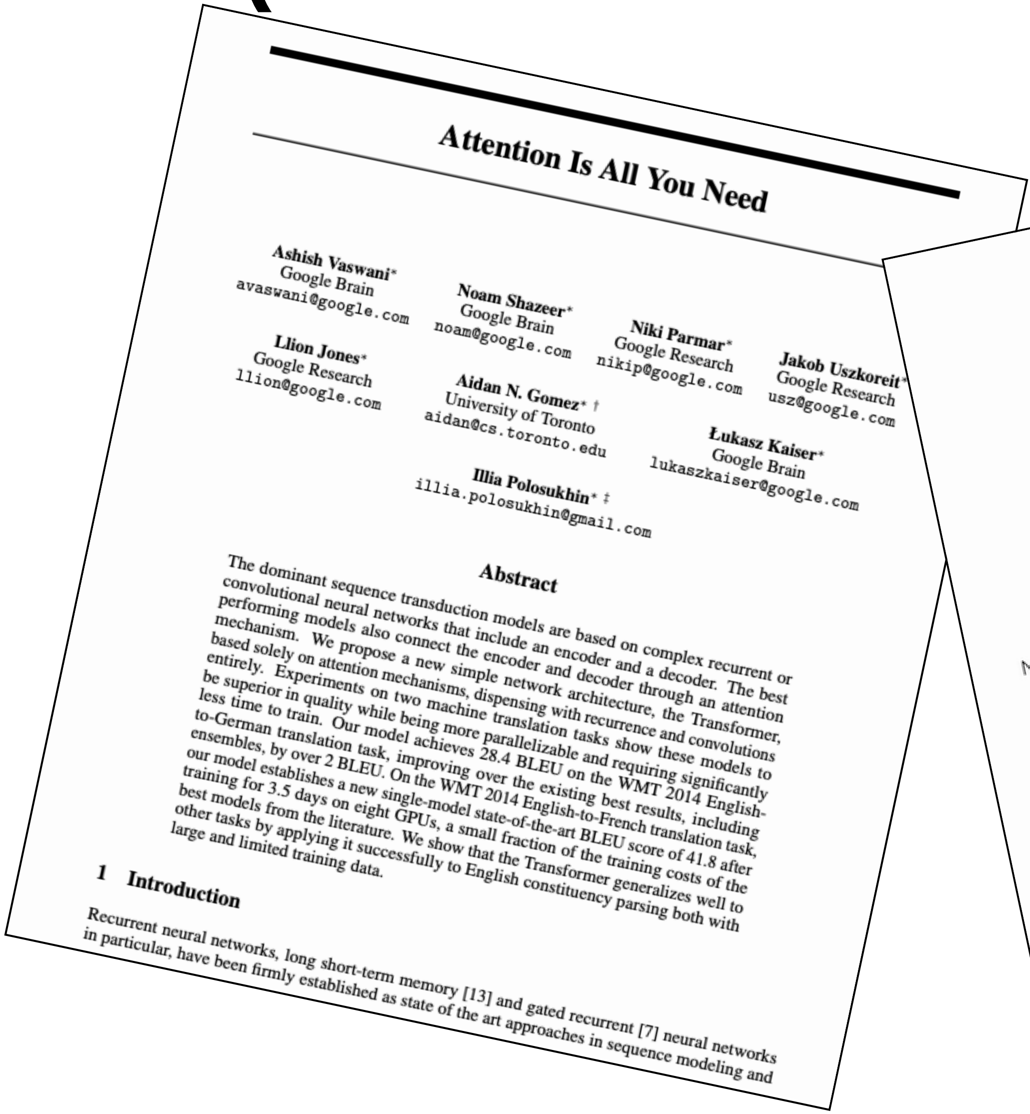
fairseq

convolutional network based encoder

transformer

self-attention network based encoder

Transformer
(Vaswani et al. 2017)


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
lukaszkaiser@google.com

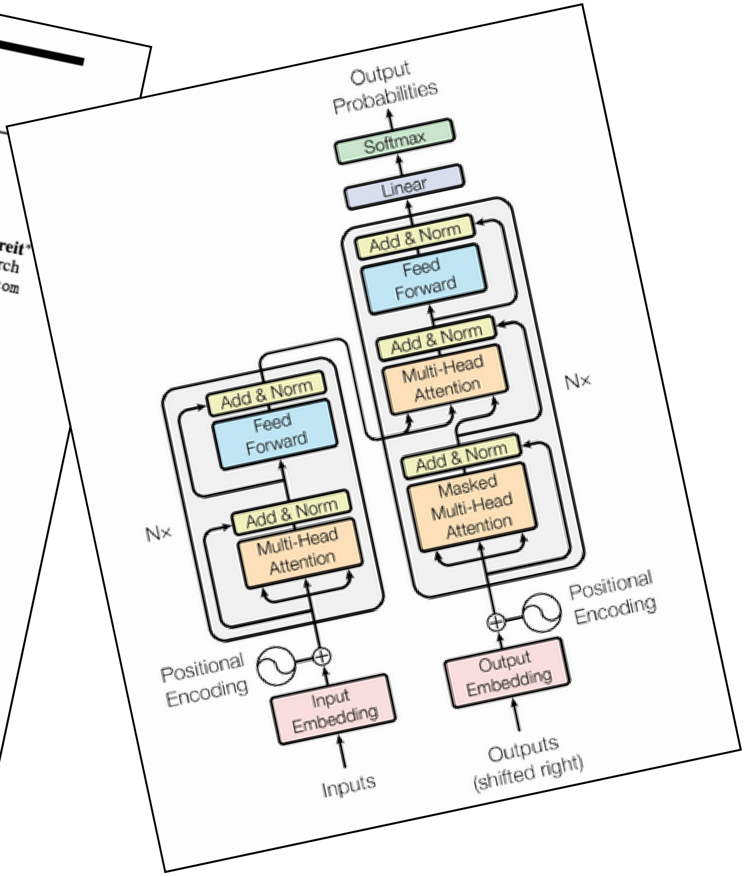
Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

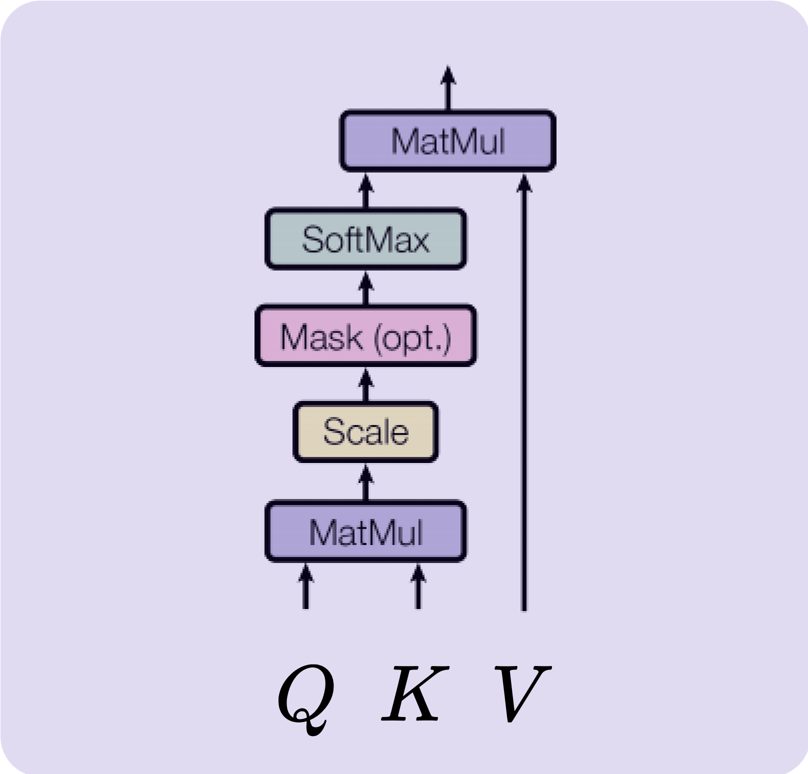
1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and



Scaled Dot Product

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{\delta}})V$$

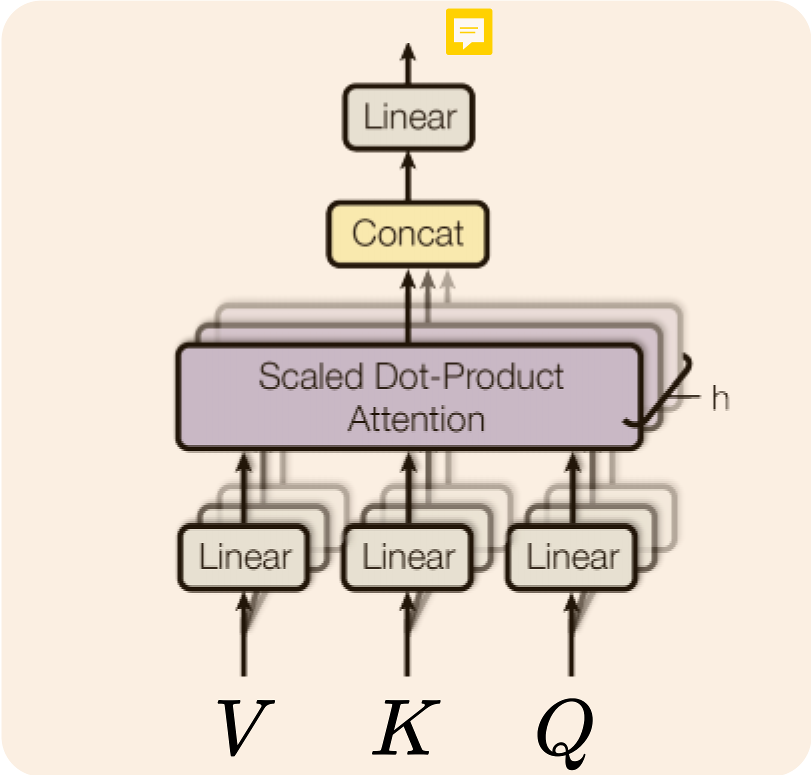


Multi-Head Attention

Multiple heads for attending to different information

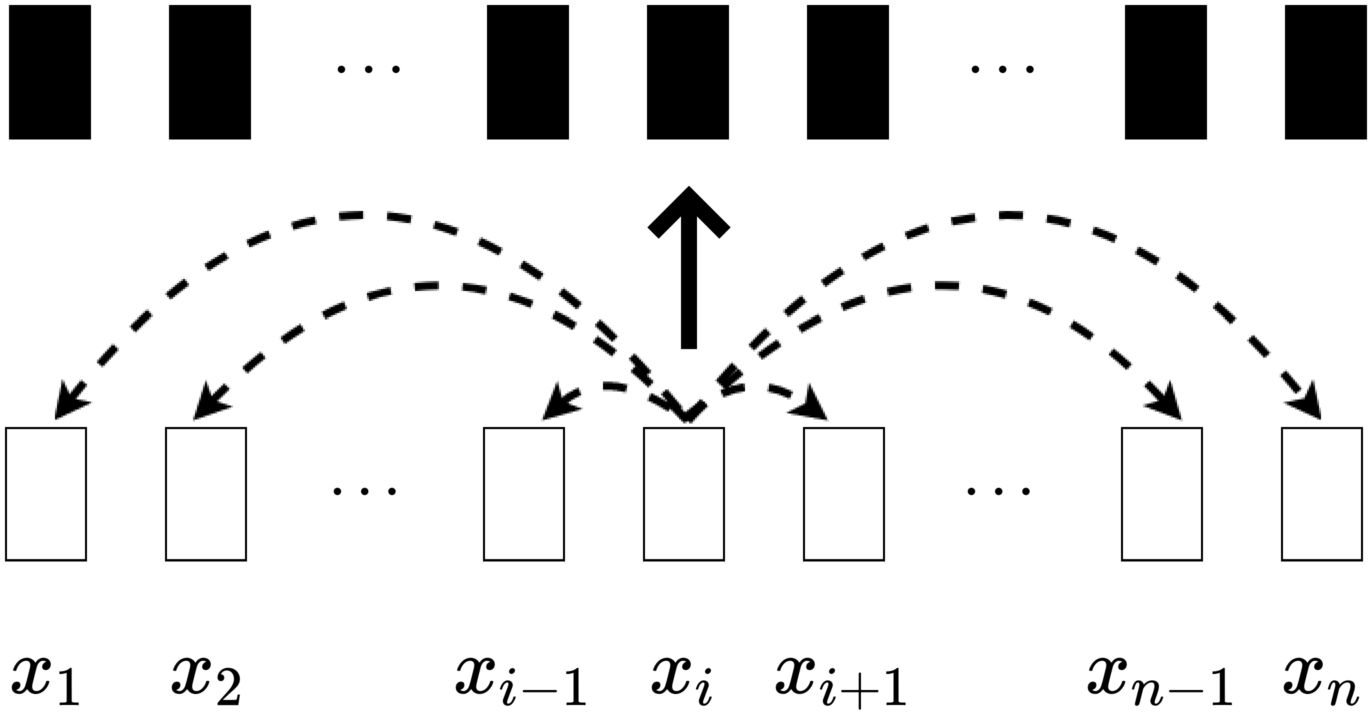
$$\text{MultiHead}(Q, K, V) = [\text{head}_1; \text{head}_2; \ldots ; \text{head}_h]W^O$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



Note that the right most is now Q!

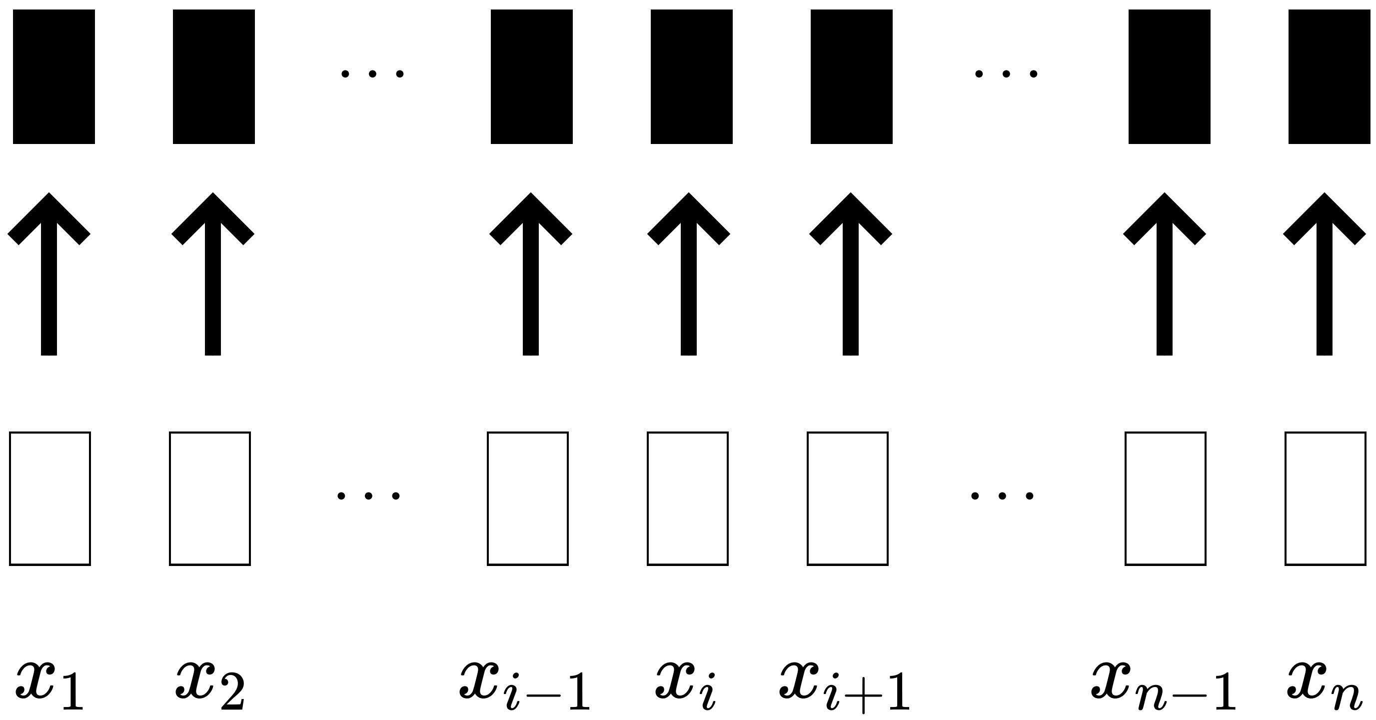
Self Attention



$$Q = K = V$$

Query, key and value matrices are the same!

Self Attention

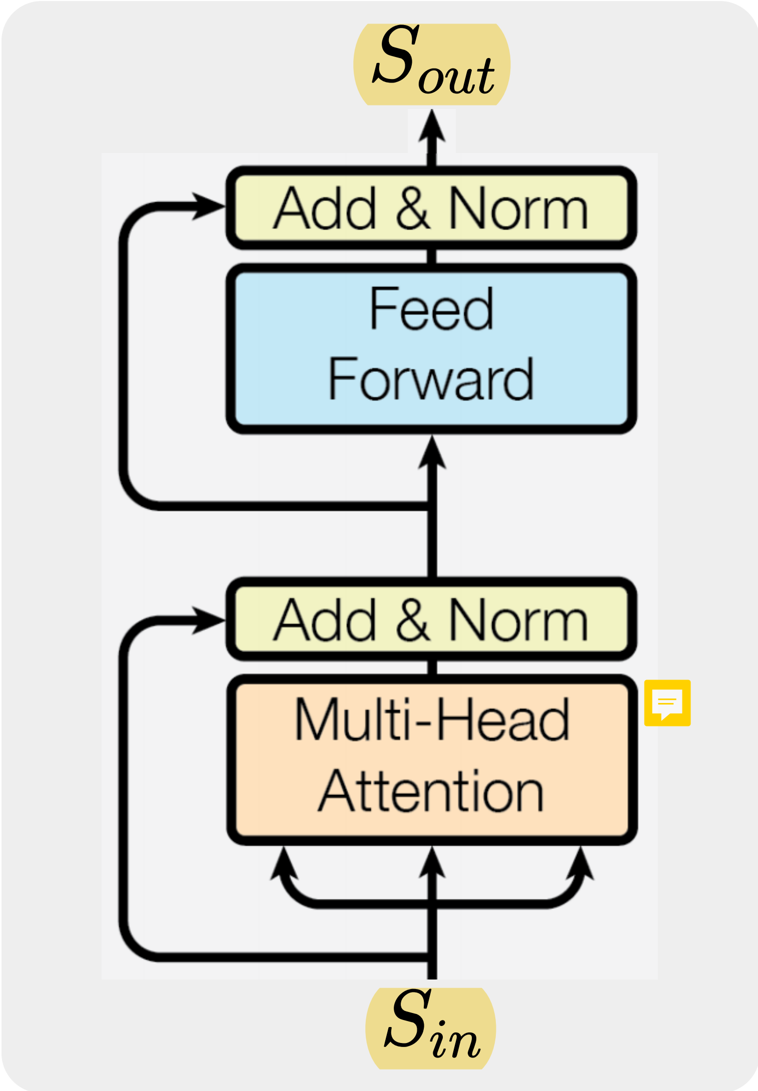


$$Q = K = V$$

Captures "self-similarity" or relations between elements in a complete sentence. Can well capture long-range dependencies, but does not require recurrent structures.

Encoder

source context embedding



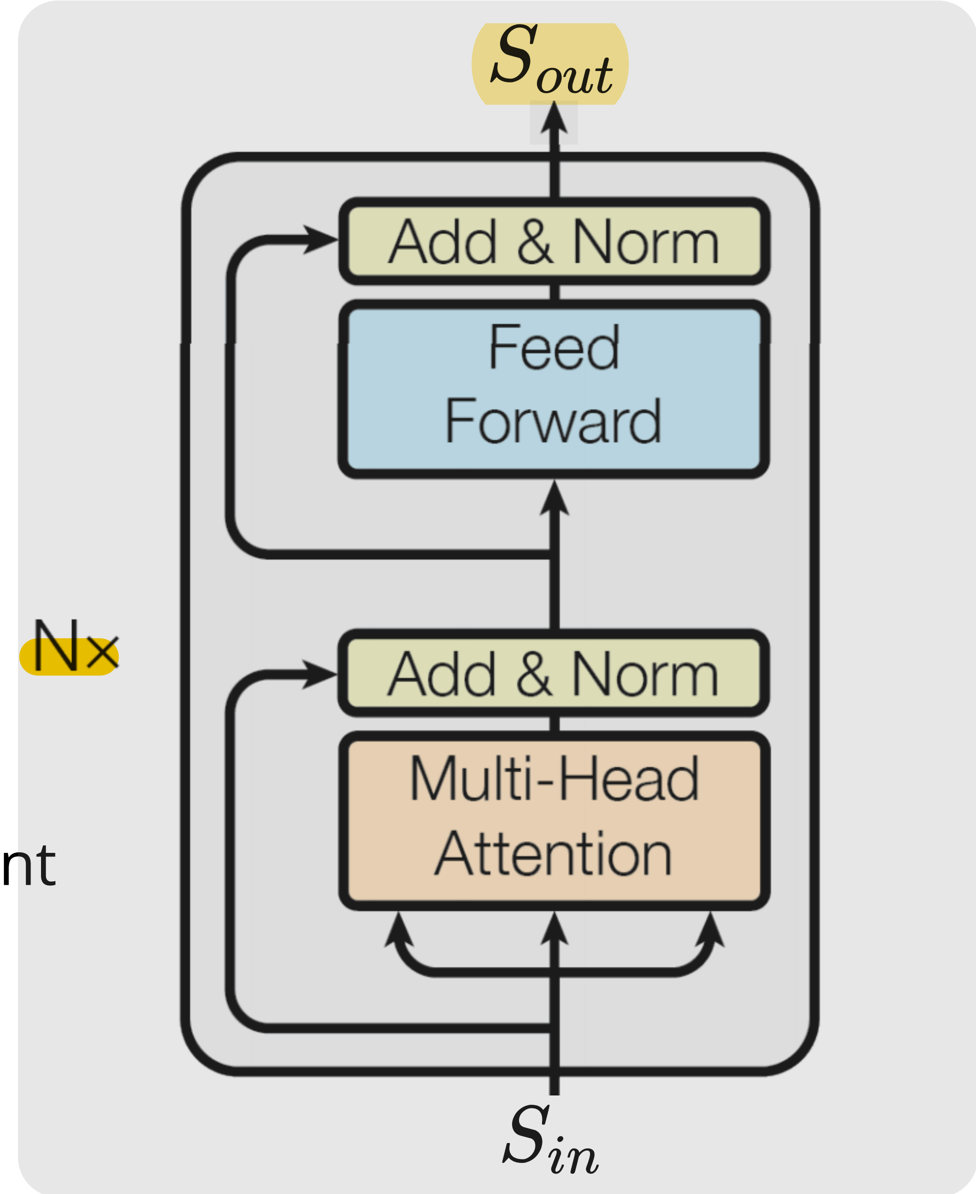
Position-wise feedforward network

Highway connection with layer normalization

source input embedding

Encoder

(deep) source context embedding

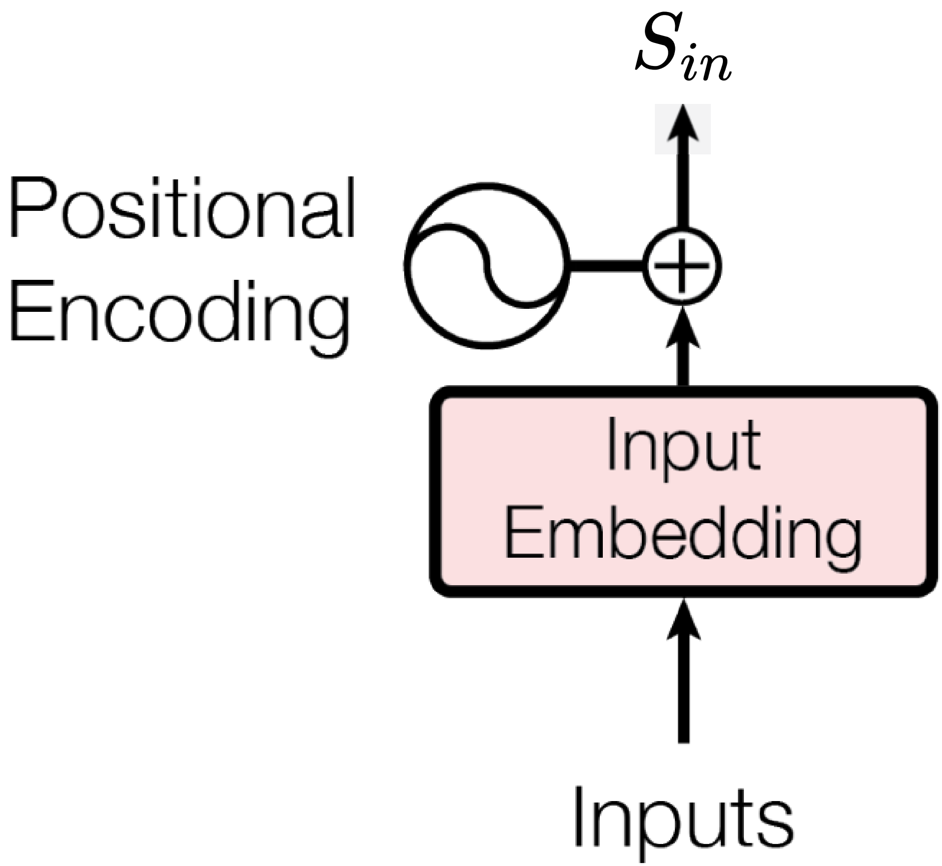


Stack N of such blocks!

How to represent the input?

source input embedding

Encoder



The input embeddings are augmented with positional encoding

Optional

Positional Encoding

No recurrence and no convolution

Some information about the relative or absolute position of the tokens in the sequence must be injected

position in the sentence

dimension in the embedding

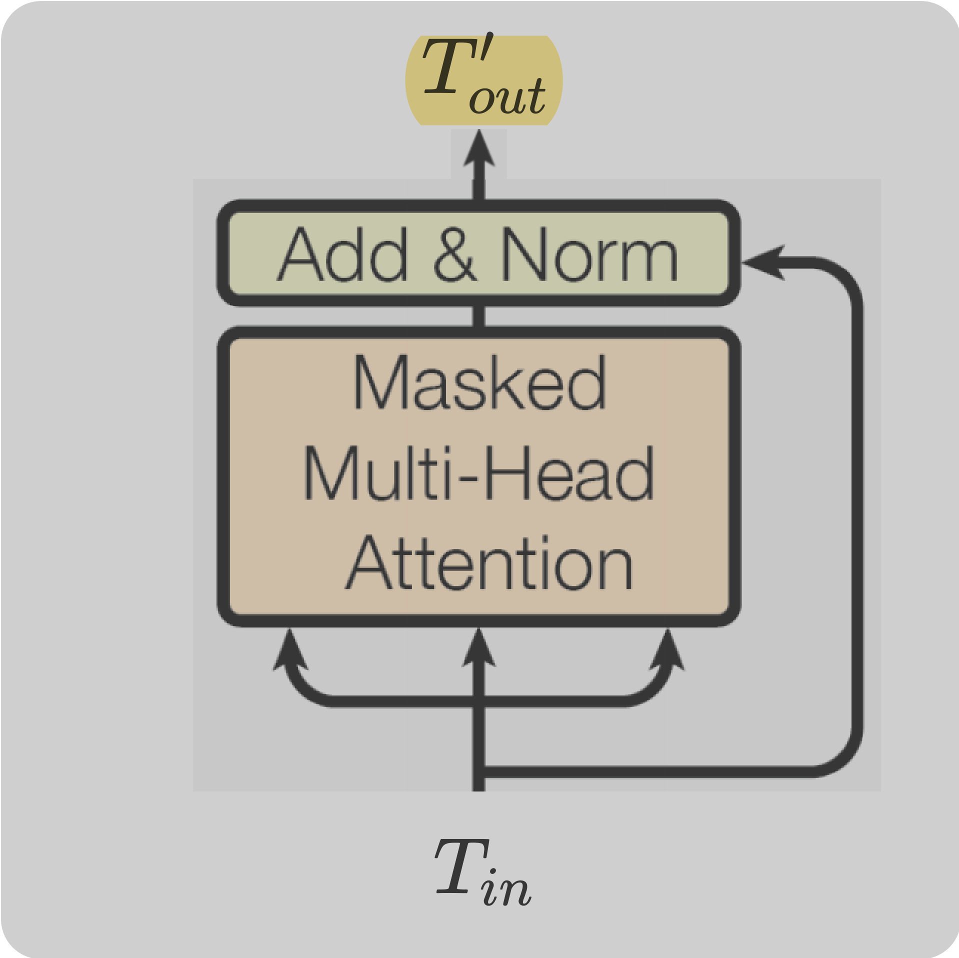
$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}})$$

Similar results may be obtained by learning "position embeddings"

Decoder

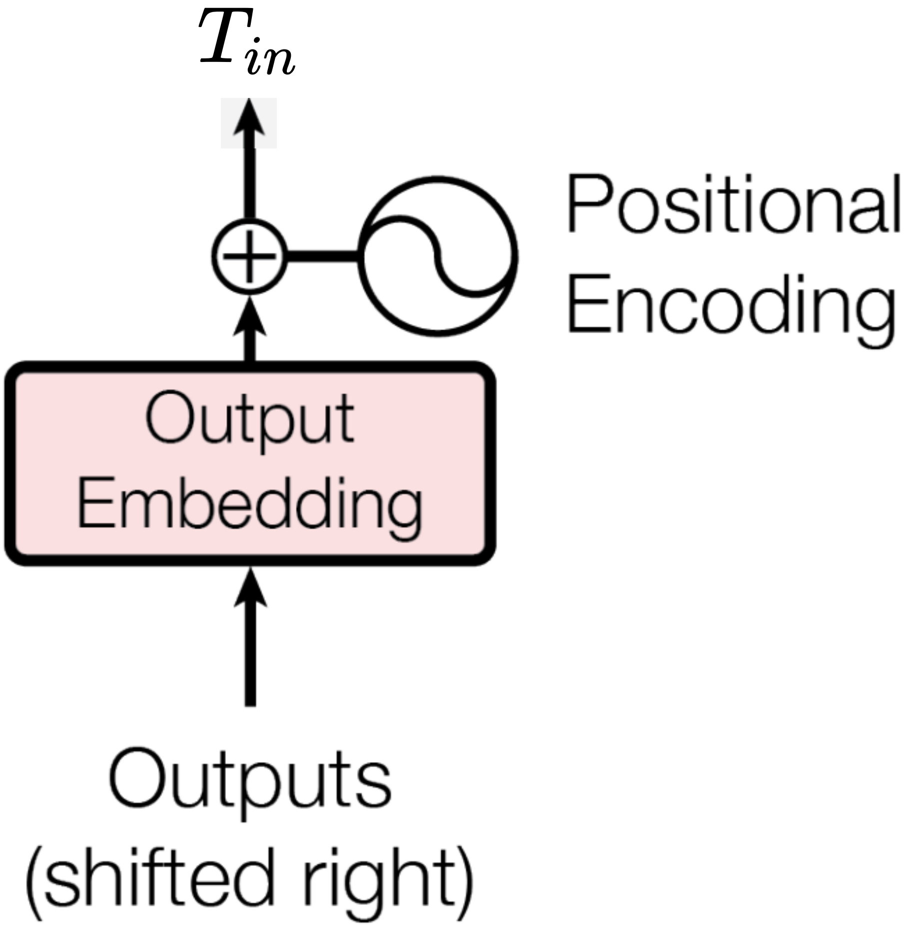
target context embedding



target input embedding

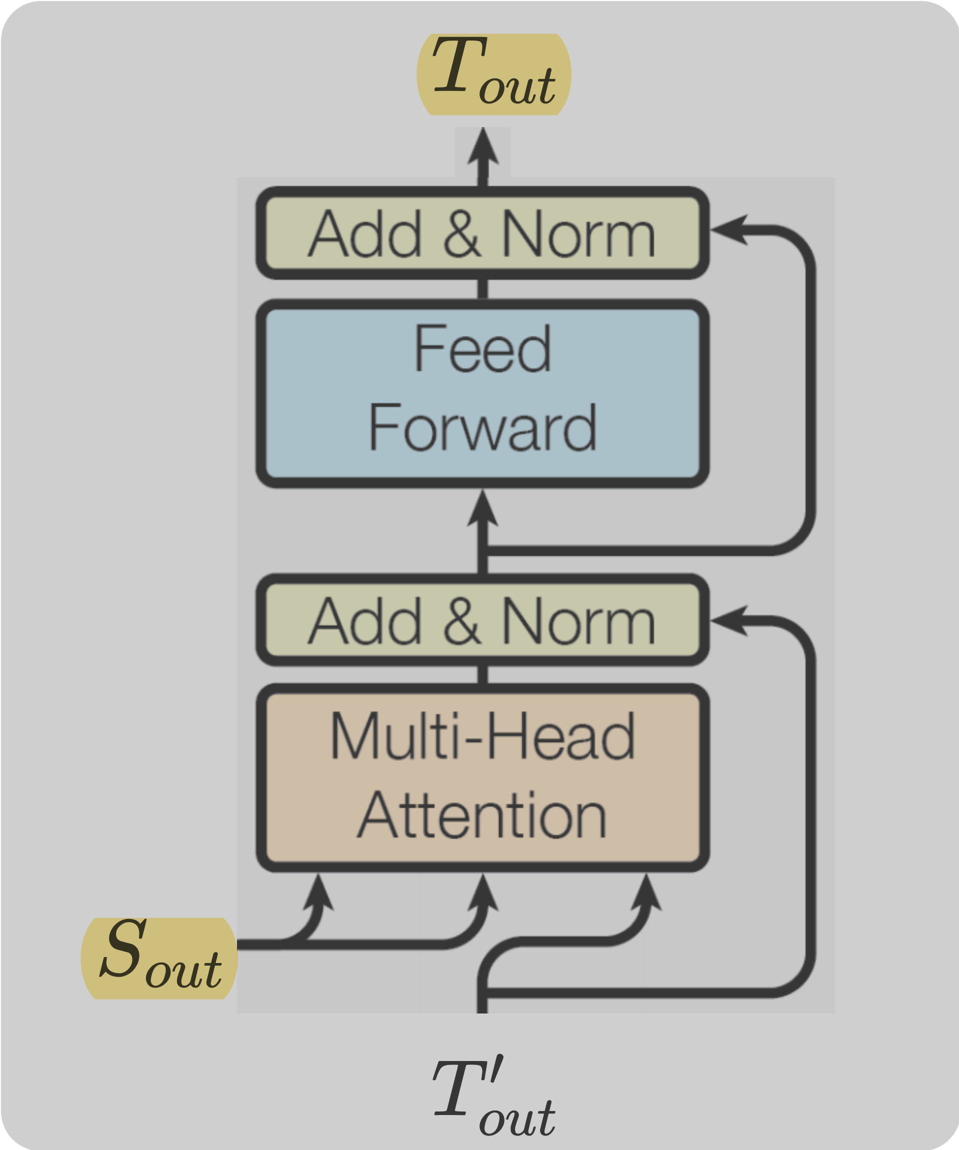
As at each step, we only have a partial output sequence, we will use Masked attention

Decoder



Similar architecture for the decoder input. Note that the outputs grows longer from left to the right

Decoder

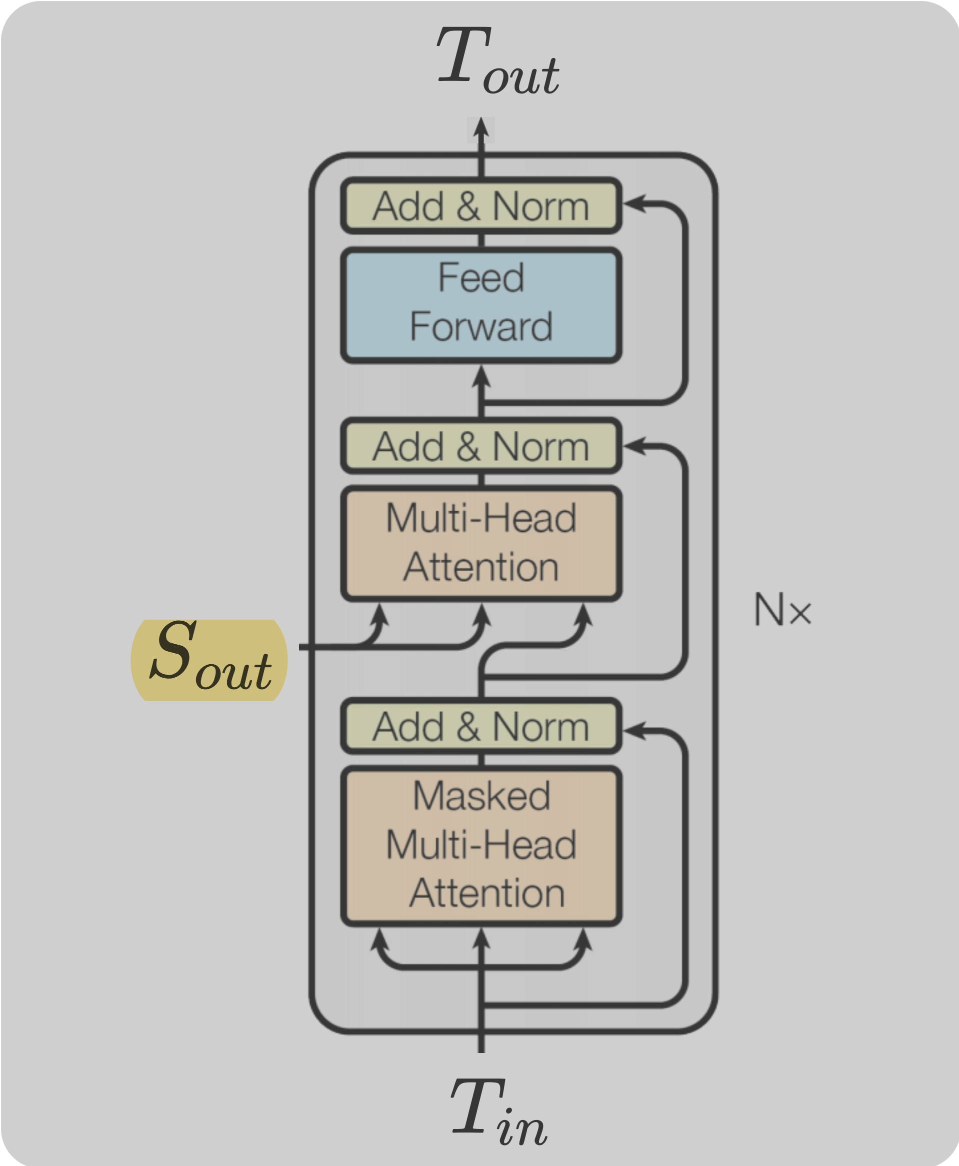


This is the retrieved content

This is the key and value

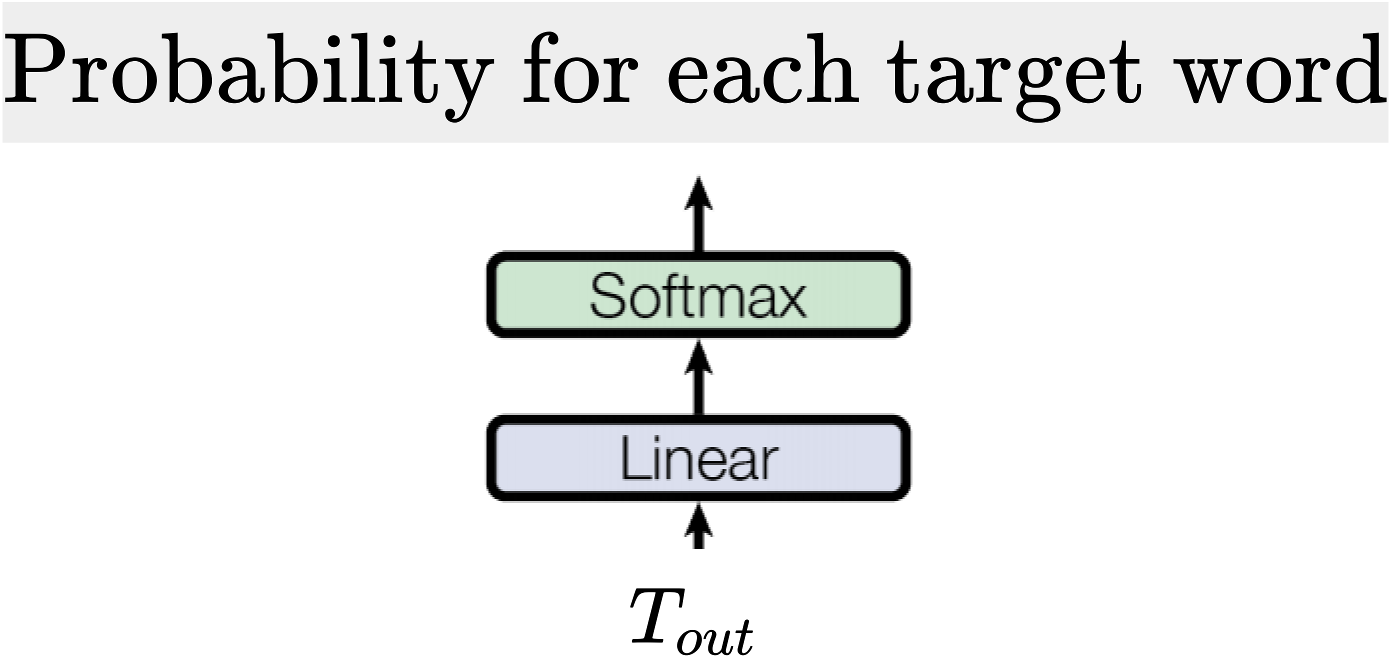
This is the query

Decoder



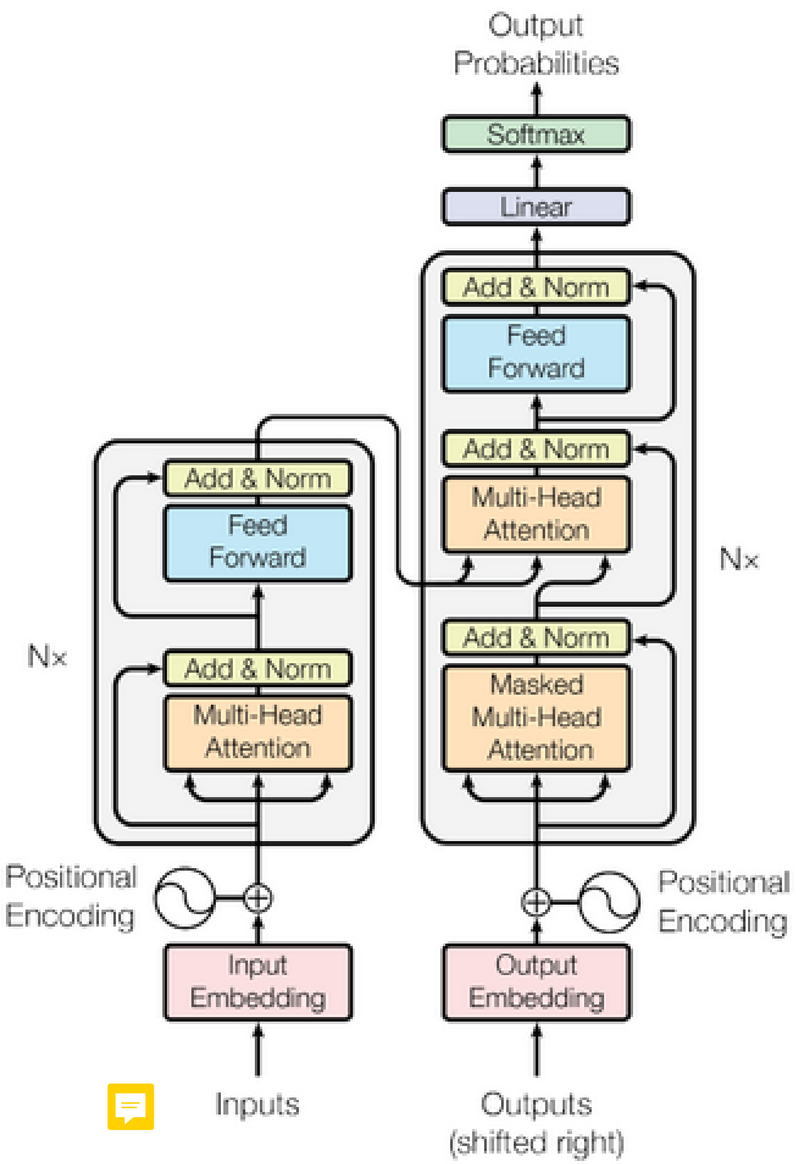
Stack N of such blocks!

Decoder



The training procedure would be similar to that of the seq2seq

Transformer



Encoder

Decoder

Comparison

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 d)$	$O(1)$	$O(1)$
Recurrent	$O(nd^2)$	$O(n)$	$O(n)$
Convolutional	$O(knd^2)$	$O(1)$	$O(\log_k n)$
Self-Attention (restricted)	$O(rnd)$	$O(1)$	$O(n/r)$

n is the sequence length

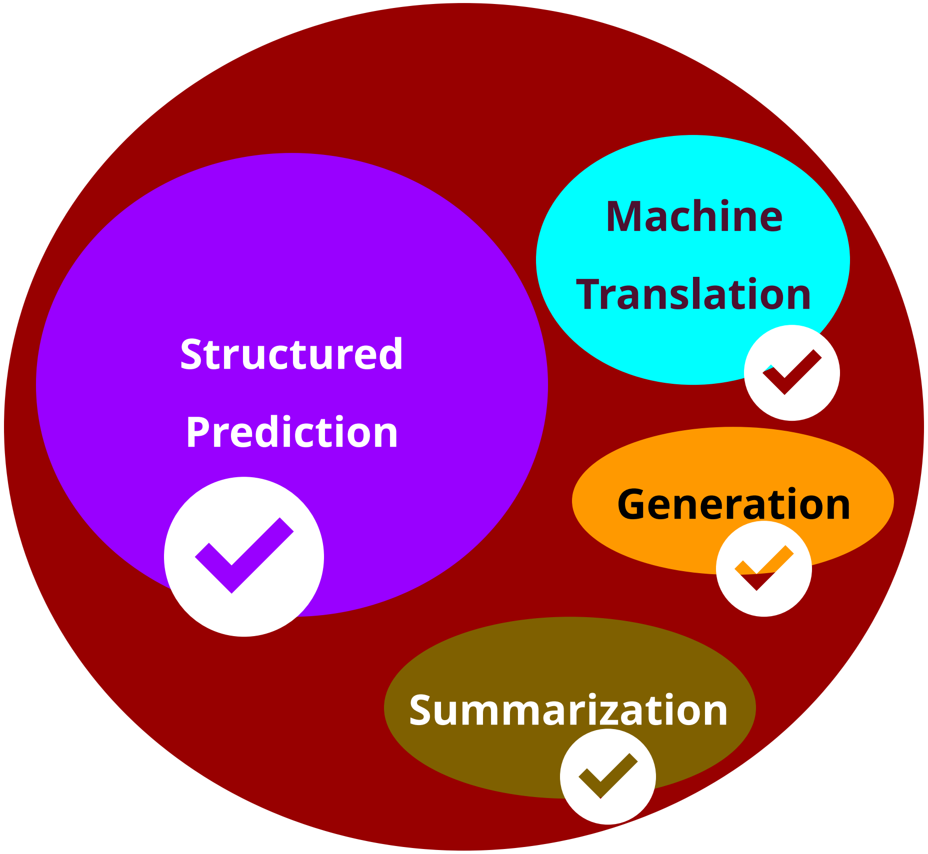
d is the representation dimension

k is the kernel size of convolutions

r is the size of the neighborhood in restricted self-attention

maximum path length tells us how easy the model is to capture long-range dependencies. The lower the better

Tasks in NLP



Supervised

Tasks in NLP

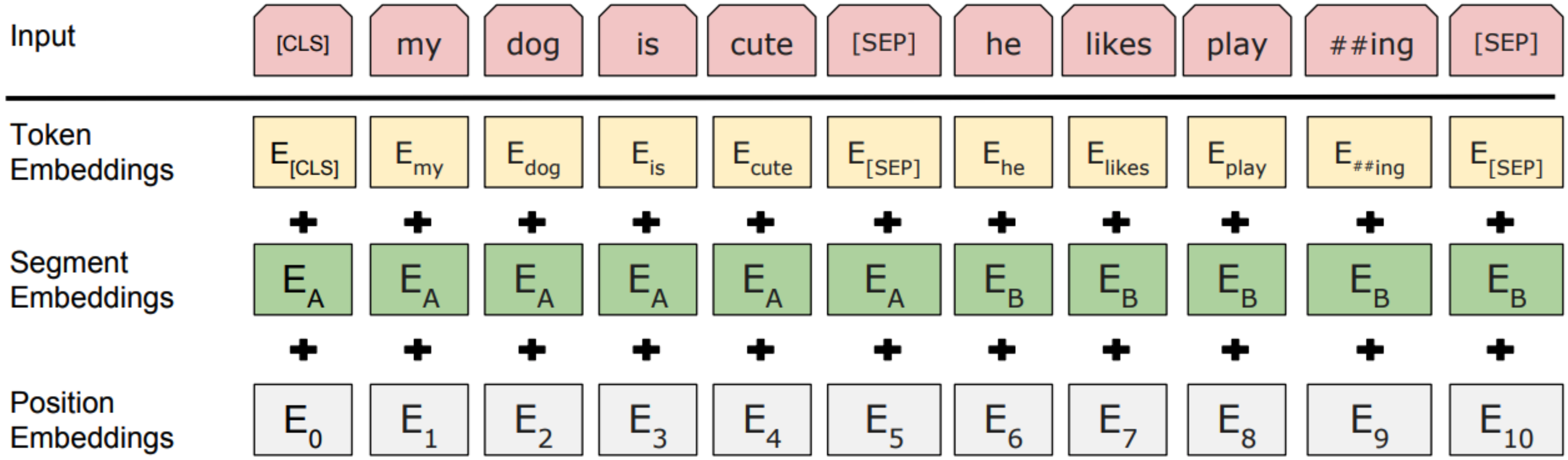
POS Tagging
Chunking
Document Classification
Information Extraction
Syntactic Parsing
Semantic Parsing
Natural Language Generation
Machine Translation
Sentiment Analysis
Coreference Resolution
Question Answering

Supervised

Word Clusters
GloVe, word2vec
Topic Modeling
Language Modeling
ELMo, BERT, …

Unsupervised

BERT
(Devlin et al. 2018)



The input embeddings are the sum of the token embeddings, the segmentation embeddings and the position embeddings.

[CLS]: the starting state

[SEP]: the end of a sentence

BERT

Two Pre-training Tasks

Task #1: Masked LM

Useful for learning context representations within a sequence

Task #2: Next Sentence Prediction

Useful for tasks that involve identifying relations between multiple sentences

Masked LM
Pre-training Task #1

the man went to the store . he bought a gallon of milk .

Randomly replace 15% of the words as [MASK]

the man went to the [MASK] . he bought a [MASK] of milk .

Train the Transformer Encoder such that its learned context embeddings at the specific positions can be used to predict the masked words

Next Sentence Prediction

Pre-training Task #2

A : The man went to the store .

B : He bought a gallon of milk .

Label: IsNextSentece

A : The man went to the store .

B : penguins are flightless .

Label: NotNextSentece

Train the Transformer Encoder such that its learned [CLS] representation can be used for predicting the label

BERT

It achieves the state-of-the-art results when used in some down-stream supervised NLP tasks.

Note that when using BERT in practice, some fine-tunings may be required in different tasks (similar to ELMo). The authors provided some guidance on this in their paper, but the process is generally inexpensive.

Tasks in NLP

POS Tagging
Chunking
Document Classification
Information Extraction
Syntactic Parsing
Semantic Parsing
Natural Language Generation
Machine Translation
Sentiment Analysis
Coreference Resolution
Question Answering

Supervised

Word Clusters
GloVe, word2vec
Topic Modeling
Language Modeling
ELMo, BERT, ...

Unsupervised