Attention Is All You Need

The Rise of the Transformer

Jong-Cheol Lee

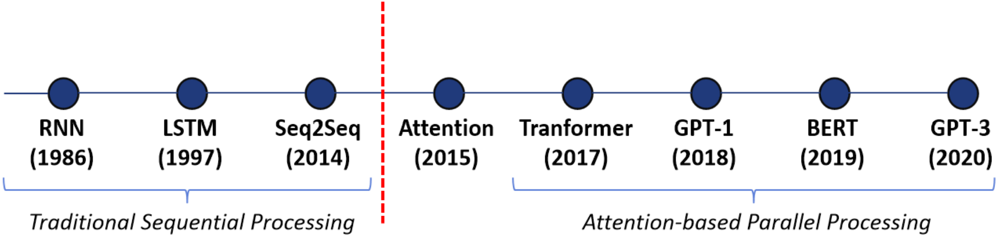
Department of Statistics
Pusan National University

October 29, 2024

Contents

Introduction

Machine Translation



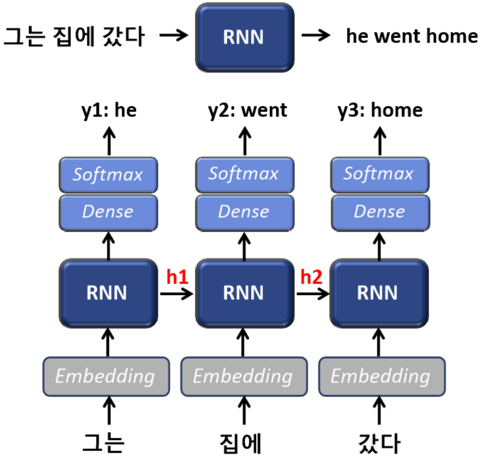
Goal

Key Objective

"Proposing the **Transformer**, *a completely new architecture."*

RNN-based Translation

Example



Fundamental Concepts

- Word Embedding

$$\text{그는} \rightarrow \begin{bmatrix} 0.7 & 0.3 & \cdots & 0.4 & 0.4 \end{bmatrix}$$

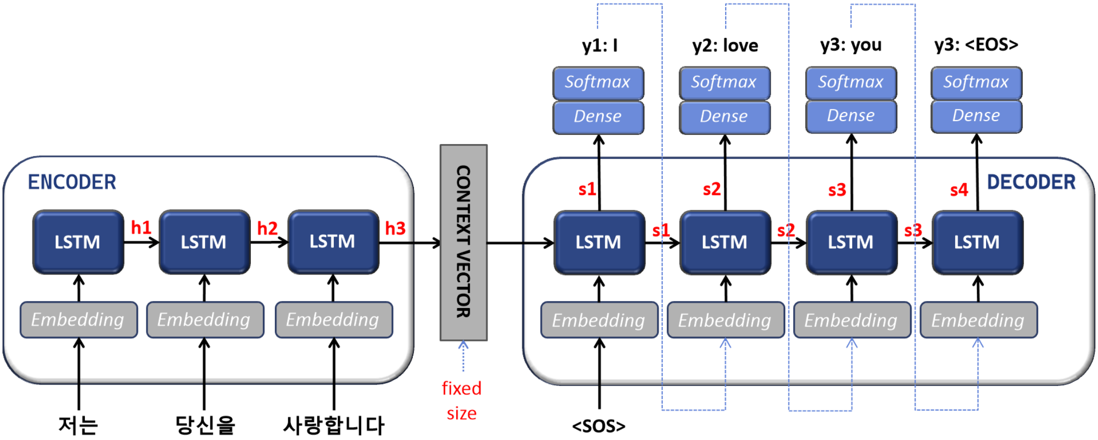
- Softmax

$$\begin{pmatrix} 2.0 \\ 1.0 \\ 0.1 \end{pmatrix} \rightarrow \frac{e^z}{e^2 + e^1 + e^{0.1}} \rightarrow \begin{pmatrix} 0.67 \\ 0.23 \\ 0.1 \end{pmatrix}$$

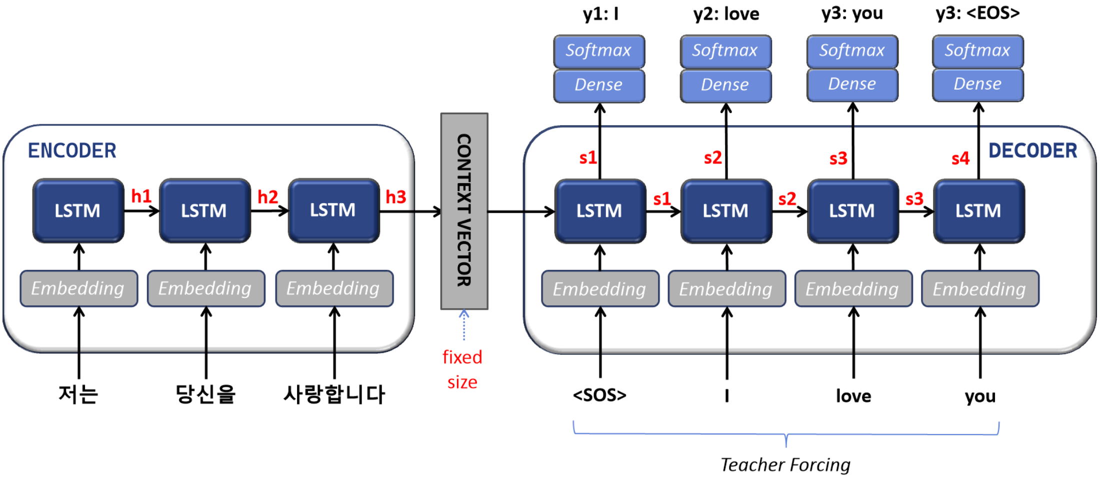
Limitations

- Input-Output Length Constraint
- Word Order Variation Across Languages

Encoder-Decoder structure



Encoder-Decoder



Attention: Background & Main Concept

Background

- Information loss of context vector
- $s_1, \ldots, s_4$: Inefficient vector representation

Assumption

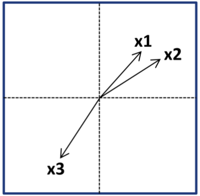
- h_i mainly captures i-th input word
- All s_i, h_i are same-sized vectors

Main Concept

Pay **attention** to important information in each step of decoder

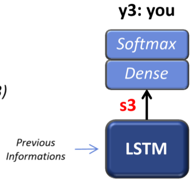
Dot-Product Attention

1) dot product ≃ similarity



dot(x1, x2) > dot(x1, x3)

2) similarity with s3 ≃ similarity with "you"



Before training: s3 is far from **"you"**

After training: s3 is close to **"you"**

3)

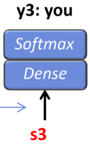
h1: Information of '저는'
h2: 〃 '당신을'
h3: 〃 '사랑합니다'

dot(h1, s3): *similarity between* **'저는'** *and* **s3**

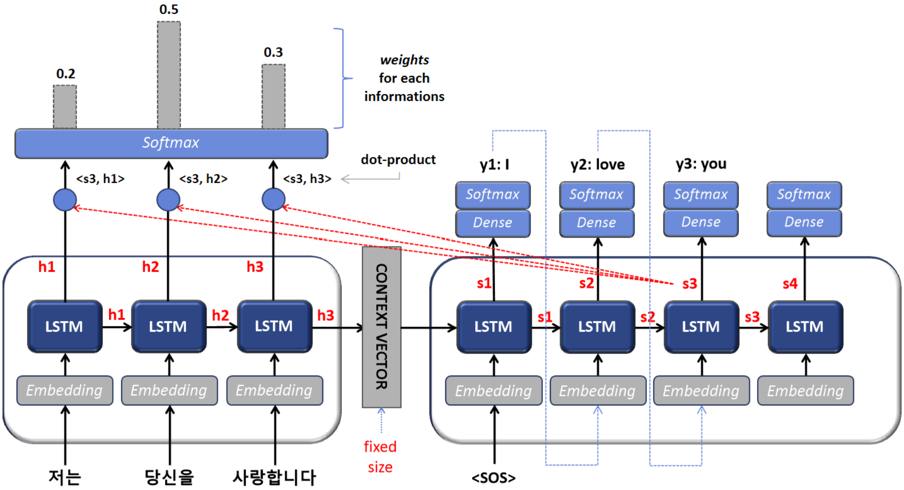
dot(h2, s3): *similarity between* **'당신을'** *and* **s3**

dot(h3, s3): *similarity between* **'사랑합니다'** *and* **s3**

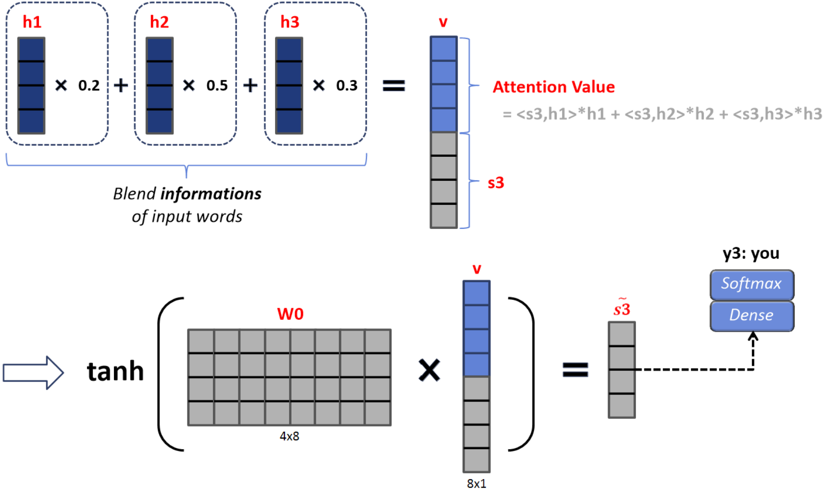
$$\underbrace{\text{dot}(h1, s3)\cdot h1 + \text{dot}(h2, s3)\cdot h2 + \text{dot}(h3, s3)\cdot h3}_{\textit{weighted sum}} = \textbf{Attention value vector}$$



Dot-Product Attention

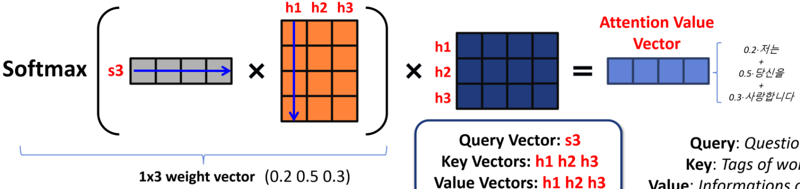


Dot-Product Attention



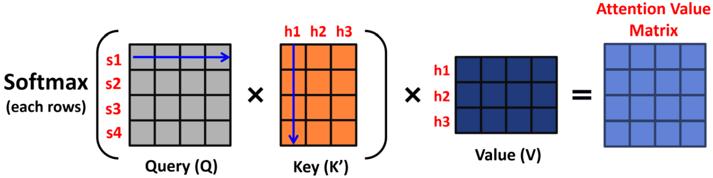
Main Idea: Query, Key, Value

What is Q, K, V?



Query Vector: s3
Key Vectors: h1 h2 h3
Value Vectors: h1 h2 h3

Query: *Questions*
Key: *Tags of words*
Value: *Informations of words*

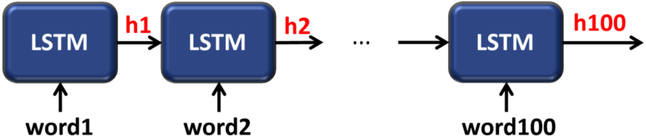


∴ Attention(Q, K, V) = softmax(QK')V

Transformer: Background & Main Concept

Problems of Sequential Processing

- Difficulty to parallelize
- Long-term dependency



Main Concept

Using only **attention mechanisms** to build an encoder-decoder

Transformer: Overview

Model Architecture

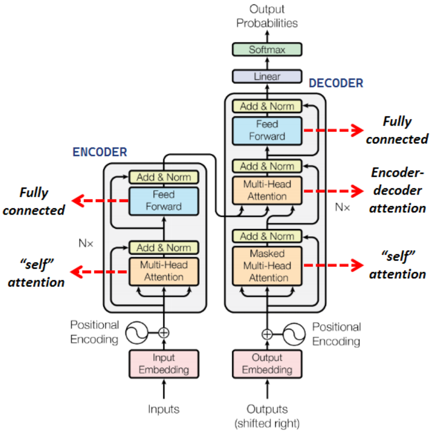


Figure 1: The Transformer - model architecture.

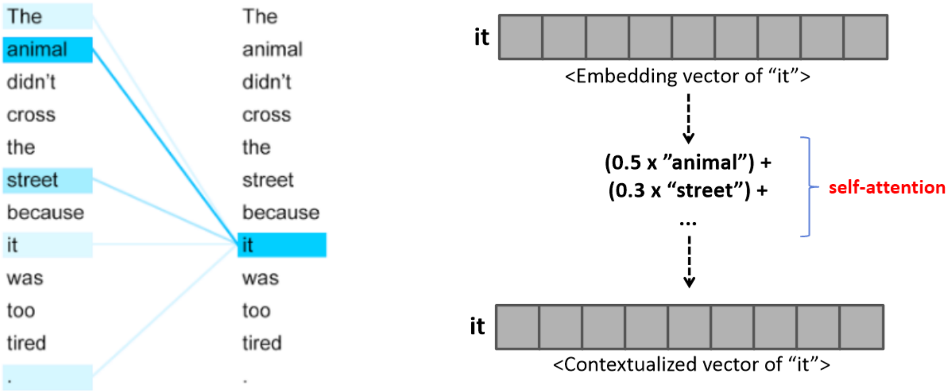
Hyper Parameters

Hyperparameter	Value
d_{model}	512
d_{ff}	2048
N_{heads}	8
N_{layers}	6

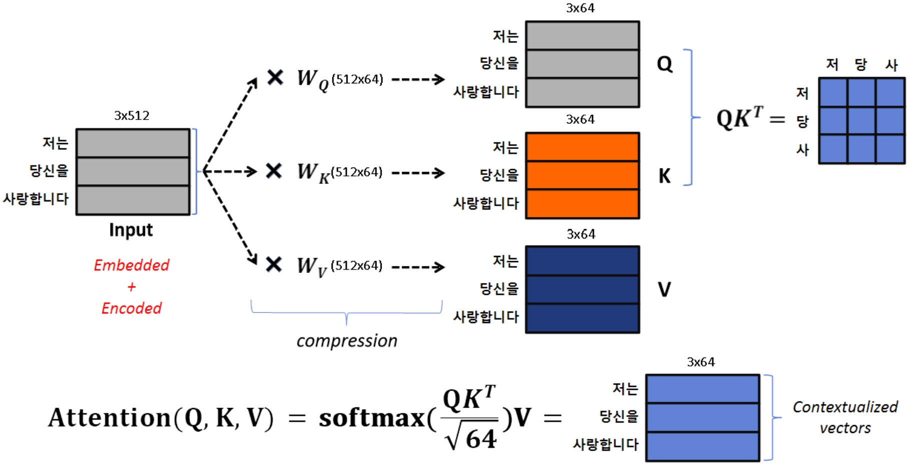
- d_{model}: size of embedded vector (equals with size input and output)
- d_{ff}: hidden size of *Feed Forward*
- N_{heads}: # of attention heads
- N_{layers}: of layers stacked

An Effect of Self-Attention

Inject contextual information to embedded vector



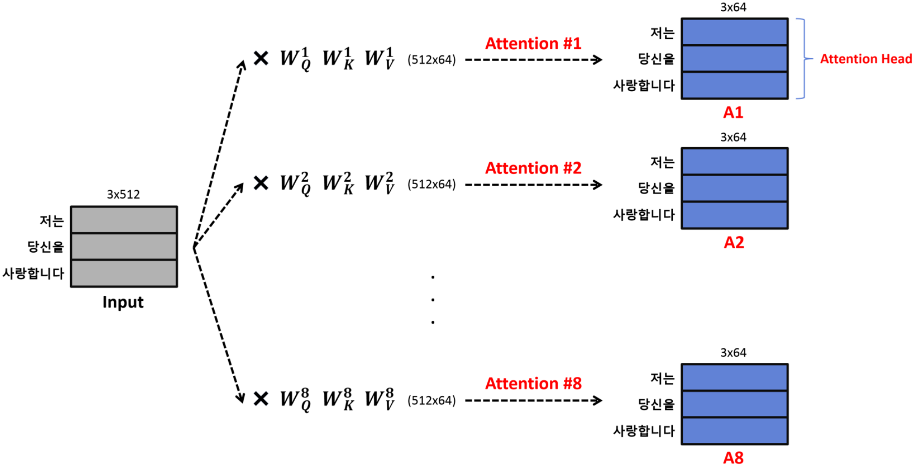
Q, K, V in Self Attention



$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{64}}\right)V =$$

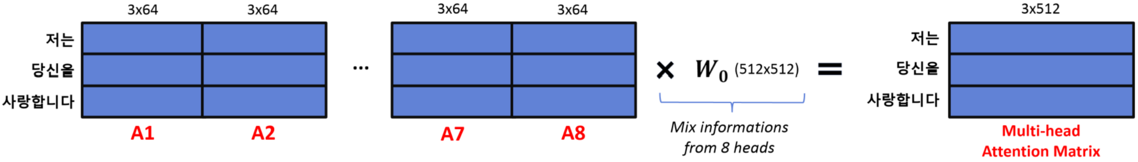
Multi-head Attention

Calculate N_{heads} Attention Matrix

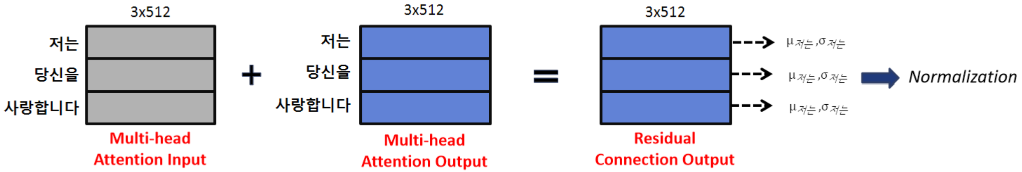


Multi-head Attention

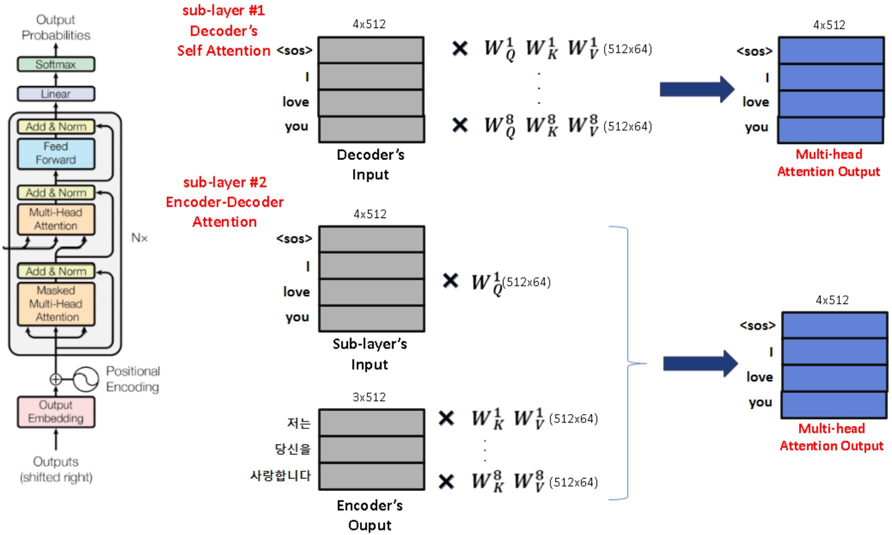
Multi-head Attention Matrix



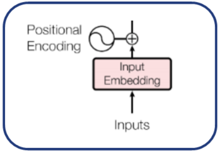
Residual Connection & Layer Normalization



Attention in Decoder

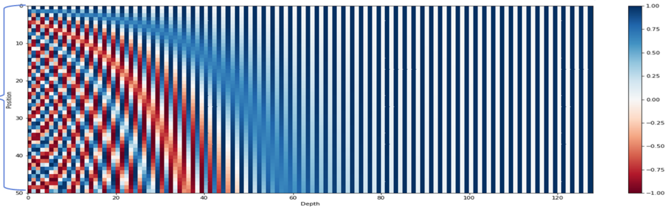


Positional Encoding



position of values
in word embedding vector
(i)

position of
each words
in sentence
(pos)



	3x512				3x512
저는		$cos(pos/10000^{2i/d_{\text{model}}})$		저는	
당신을		+ $sin(pos/10000^{2i/d_{\text{model}}})$	=	당신을	
사랑합니다		$cos(pos/10000^{2i/d_{\text{model}}})$		사랑합니다	

Input

Embedded

Input

Embedded
+
Encoded

Input matrix
with
positional information

Stacking Encoders and Decoders

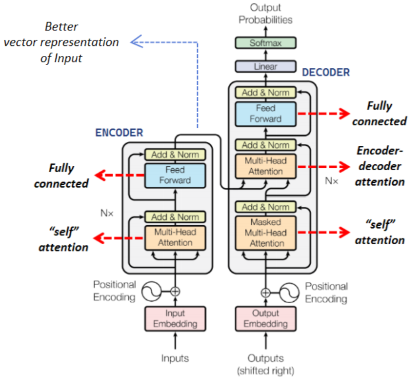
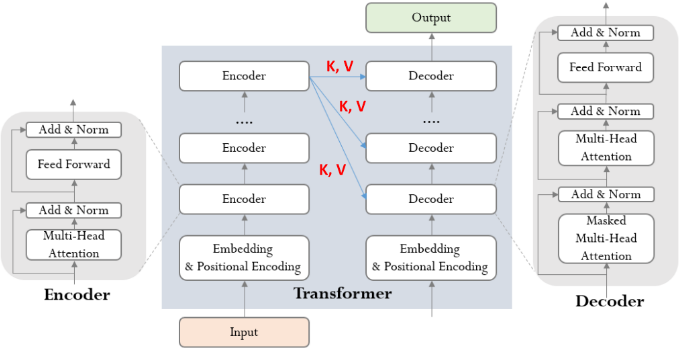
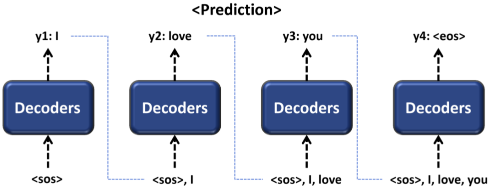
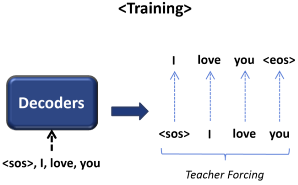


Figure 1: The Transformer - model architecture.

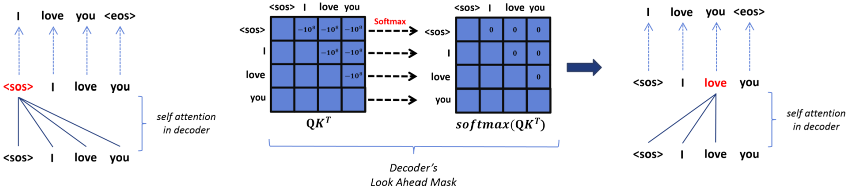


Prediction & Training





Masked Self Attention



Evaluation

Table: Comparison with English-to-German and English-to-French newstest2014 tests

Model	Base	BLEU		Training Cost (FLOPs)	
		EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	CNN	23.75	39.2	1.0×10^{20}	-
Deep-Att + PosUnk [39]	RNN + Attention	-	39.2	-	1.0×10^{20}
GNMT + RL [38]	RNN	24.6	39.92	2.3×10^{19}	1.4×10^{20}
ConvS2S [9]	CNN	25.16	40.46	9.6×10^{18}	1.5×10^{20}
MoE [32]	MoE	26.03	40.56	2.0×10^{19}	1.2×10^{20}
Deep-Att + PosUnk Ensemble [39]	RNN + Attention	-	40.4	-	8.0×10^{20}
GNMT + RL Ensemble [38]	RNN	26.30	41.16	1.8×10^{20}	1.1×10^{21}
ConvS2S Ensemble [9]	CNN	26.36	41.29	7.7×10^{19}	1.2×10^{21}
Transformer (base model)	Transformer	27.3	38.1	3.3×10^{18}	-
Transformer (big)	Transformer	**28.4**	**41.8**	2.3×10^{19}	-

References

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin.
Attention is All You Need
In *Advances in Neural Information Processing Systems*, 2017.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio.
Neural Machine Translation by Jointly Learning to Align and Translate
In *International Conference on Learning Representations (ICLR)*, 2015.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le.
Sequence to Sequence Learning with Neural Networks
In *Advances in Neural Information Processing Systems*, 2014.