28 November 2025

TRANSLATE ENGLISH TO KOREAN WITH TRANSFORMER

Minh Le

THE UNIVERSITY of ADELAIDE

CONTENTS

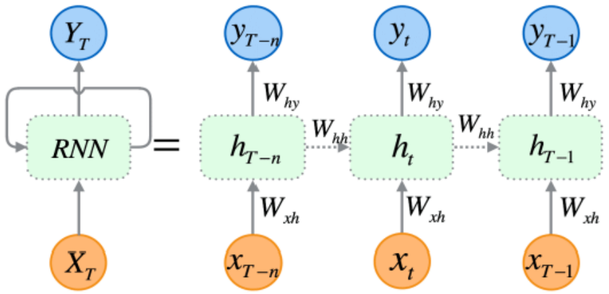
1. MOTIVATION

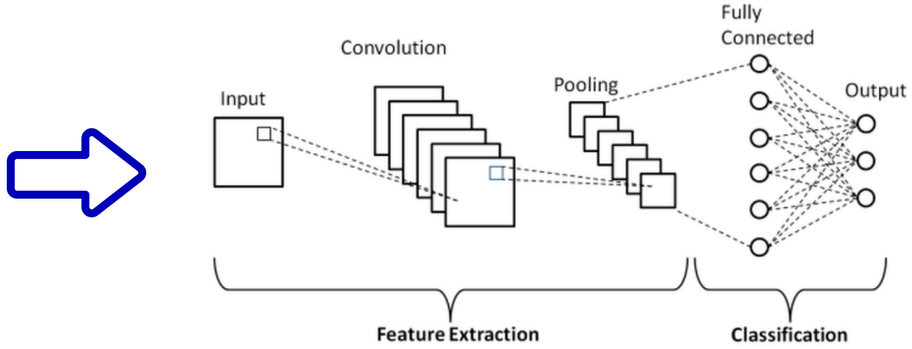
Hello, we are TWICE

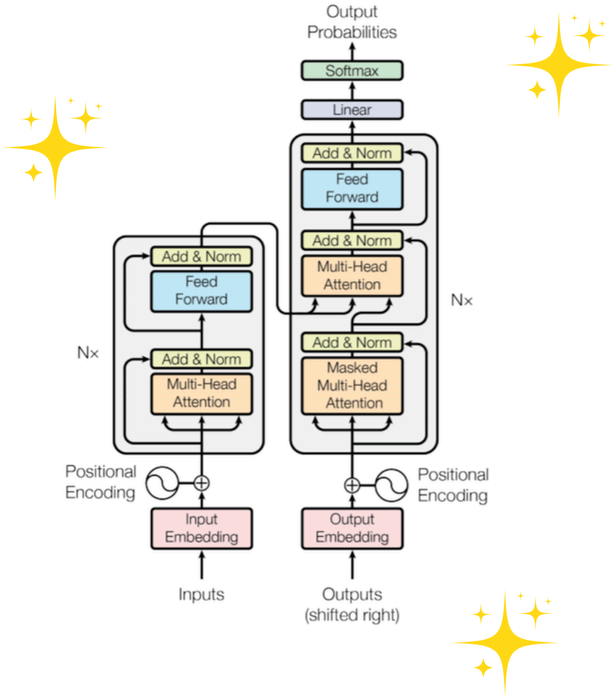
안녕하세요, 저희는 트와이스입니다

1. MOTIVATION: WHY TRANSFORMERS?

Attention Is All You Need (2017)







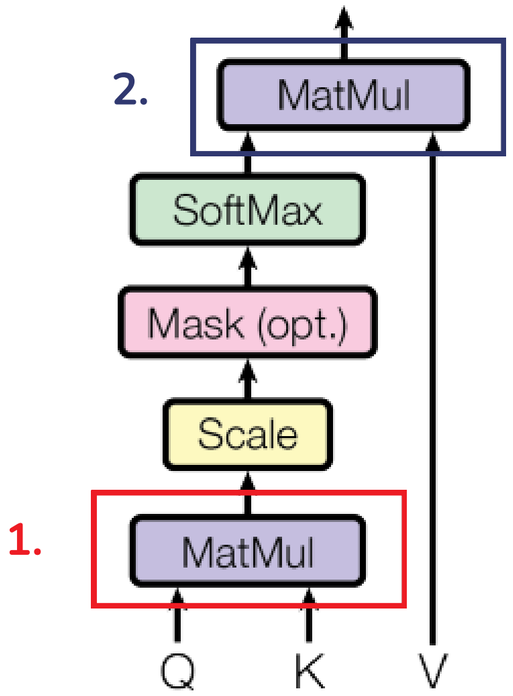
- ✗ Prone to **vanishing gradient.**
- ✗ Sequential computation **inhibits parallelization.**

- ✗ Requires many layers to capture long-term dependencies
  → **Large network**

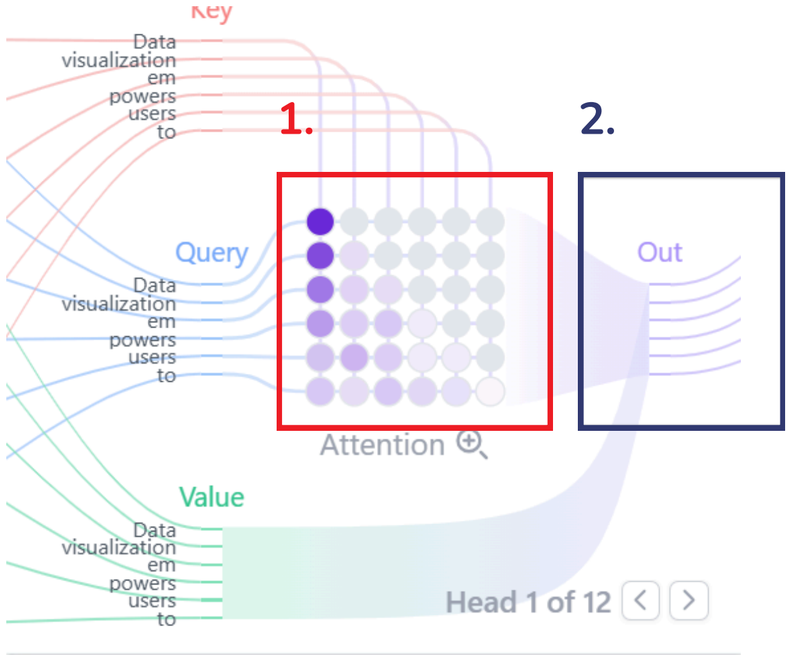
- ✓ Self-attention allows **global context.**
- ✓ All tokens go through the model simultaneously (**parallelization**)

2. TRANSFORMER ARCHITECTURE: SELF-ATTENTION

1. Compare query and key → Attention scores.
2. Apply scores to value matrix → Each token become context-aware.

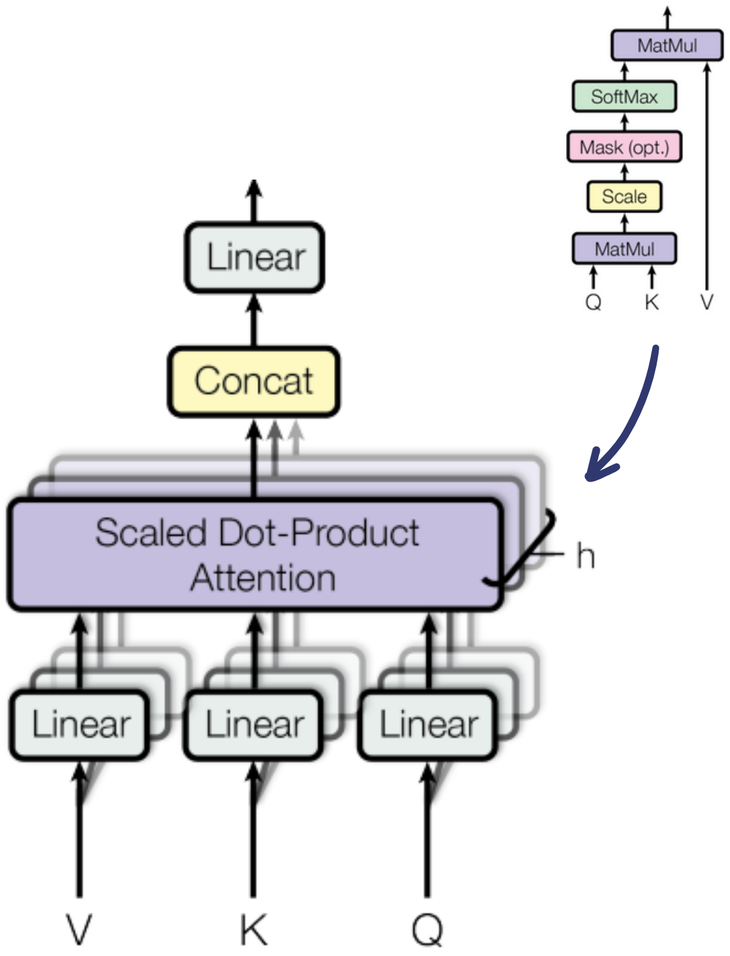


Attention Is All You Need (2017)

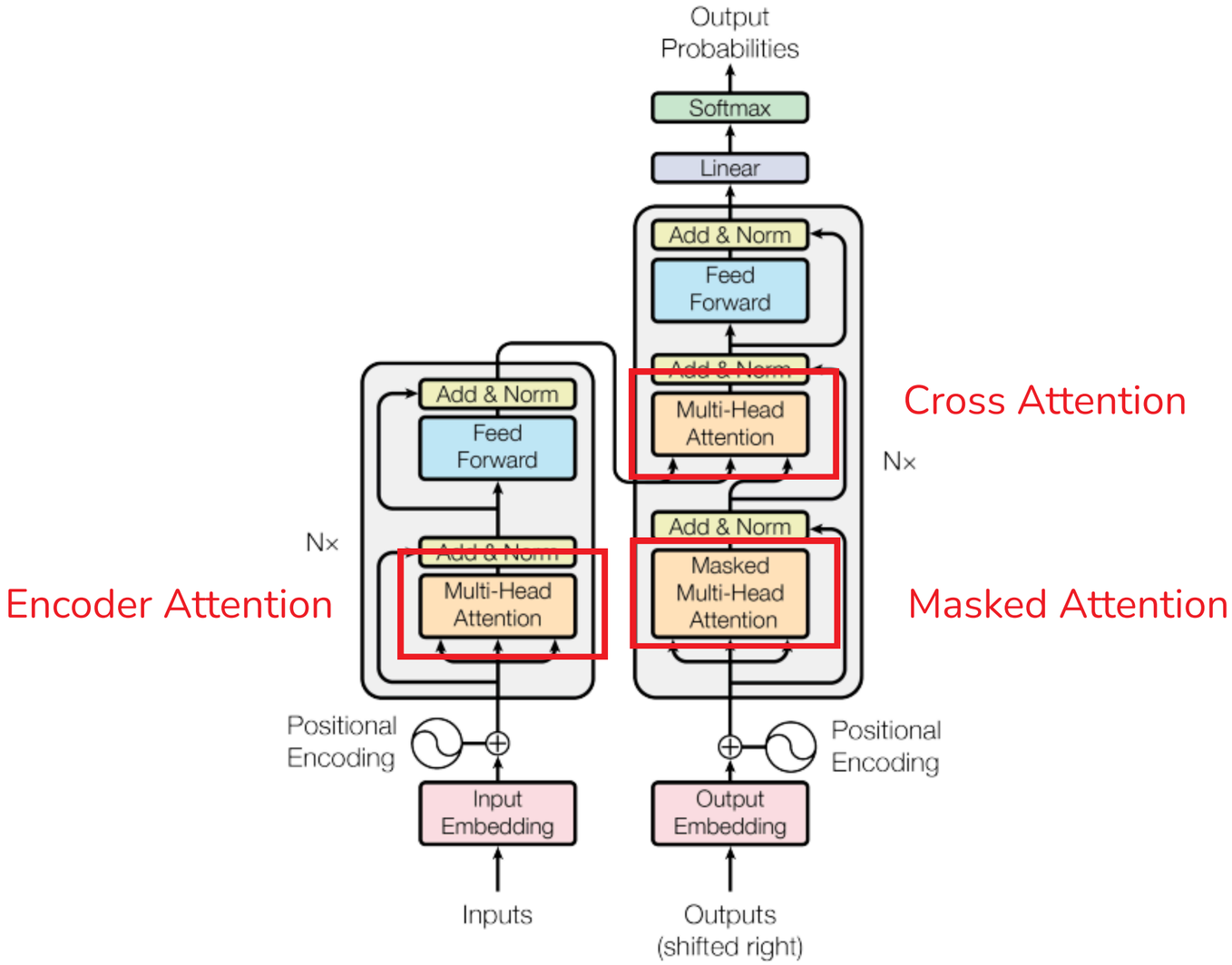


https://poloclub.github.io/transformer-explainer/

2. TRANSFORMER ARCHITECTURE: MULTI-HEAD ATTENTION



Multi-head Attention



3. IMPLEMENTATION

- I built the whole transformer **from scratch** using PyTorch, following the paper and other's projects on GitHub.

PyTorch

Attention

Generate + Code + Markdown

```python
# nn.MultiheadAttention can be used instead but for learning purposes, I built one from scratch
# https://pytorch.org/docs/stable/generated/torch.nn.MultiheadAttention.html

class MultiHeadAttention(nn.Module):
  def __init__(self, d_model, num_heads, drop_out=0.1):
    """
      d_model: the dimension of the embedding vector for EACH token
      num_heads: the number of attention heads
      drop_out: the dropout rate
    """
    super().__init__()
    assert d_model % num_heads == 0, "MultiHeadAttention.__init__(): d_model must be divisible by num_heads"
    self.d_model   = d_model
    self.num_heads = num_heads
    self.head_dim  = d_model // num_heads

    self.q_linear = nn.Linear(d_model, d_model)
    self.v_linear = nn.Linear(d_model, d_model)
    self.k_linear = nn.Linear(d_model, d_model)

    # Final linear projection after concatenating heads (Learns how to merge all heads back into a single representation)
    self.out_proj = nn.Linear(d_model, d_model)

    self.dropout = nn.Dropout(drop_out)
```

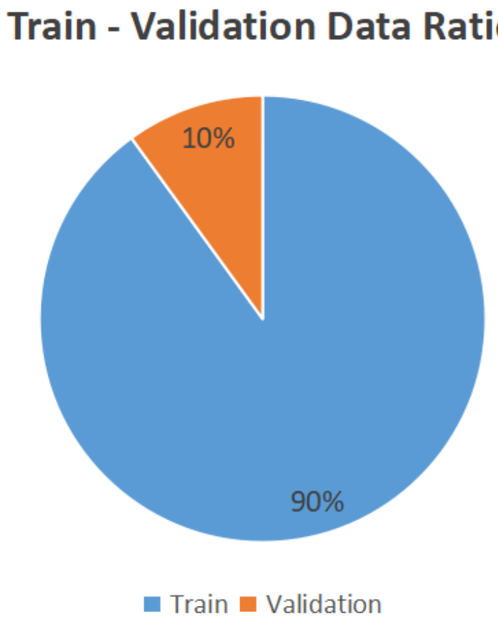
Work is still in progress...

3. IMPLEMENTATION: TOKENIZER

sentencepiece Public

https://github.com/google/sentencepiece

Dataset: **100,000 sequences (AI Hub)**



- Sequences are **reversible** (white spaces are treated as symbols).

> Hello_World.

Then, this text is segmented into small pieces, for example:

> [Hello] [_Wor] [ld] [.]

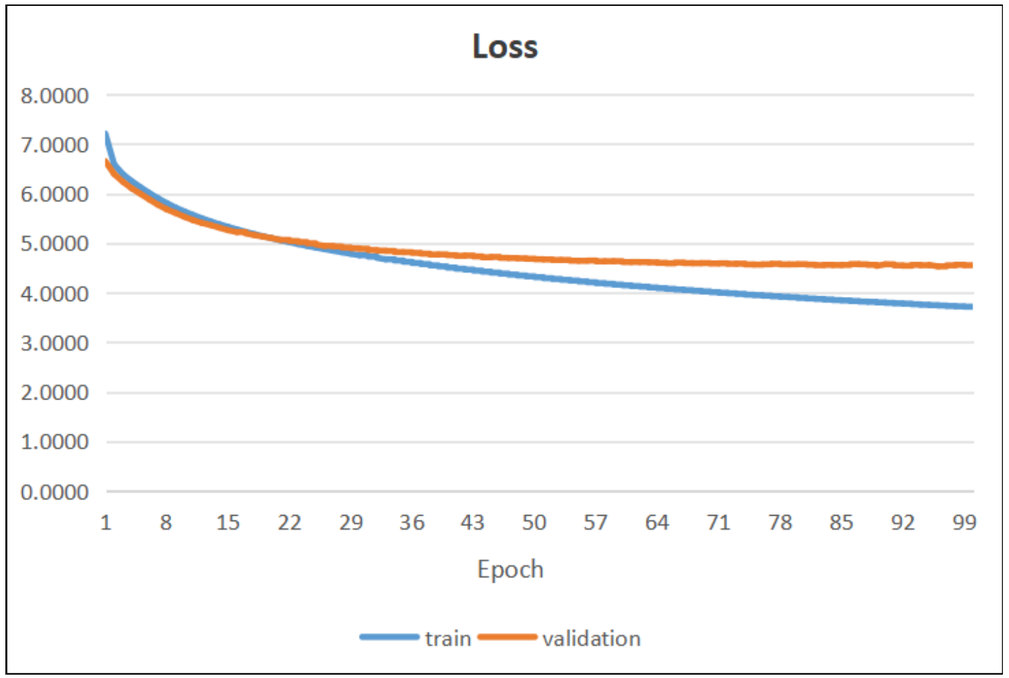
- *sentencepiece*'s unigram model is **superior to morphology.**
  - E.g. The word **obviously** can be tokenized into **_obvious** + **ly**

```
3963    _obvious    -11.3057
```

```
243    ly    -7.58103
```

This is an actual example of using *sentencepiece* on English

3. IMPLEMENTATION: RESULTS



Loss curve plateaus for the majority of the training process, suggesting that the model is **underfitting**.

- My BLEU score (ENG-KOR): **11.35**
- OPUS-MT project's BLEU score:
  - ENG-KOR: 13.3
  - KOR-ENG: 41.3

Translation:

ENG: I am going home tomorrow

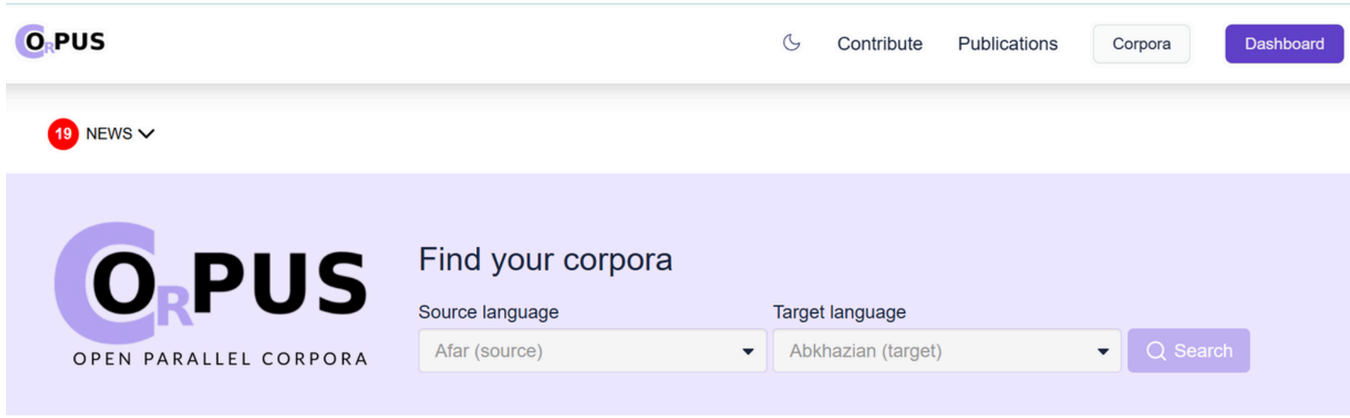
KOR: 나는 내일 집에 가는 당신과 있을 것이예요.

(I will be with you on your way home tomorrow.)

4. CONCLUSIONS: FUTURE WORK

- Apply **Beam Search**.
- **Larger dataset**.
- Fine-tune hyperparameters (embedding size, increase attention heads).

OPUS corpora: https://opus.nlpl.eu



4. CONCLUSIONS: INFLUENCE OF TRANSFORMER

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

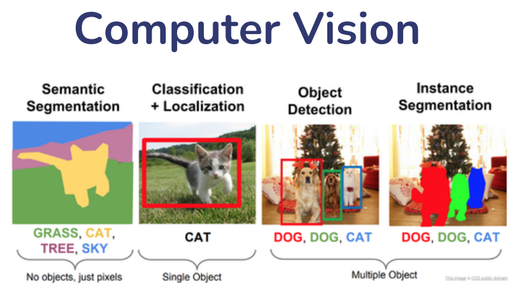
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, 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.

[*]Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

[†]Work performed while at Google Brain.

[‡]Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.



Attention Is All You Need (2017)

https://arxiv.org/pdf/1706.03762

5. REFERENCES

- Attention Is All You Need (2017) https://arxiv.org/abs/1706.03762
- Sentencepiece https://github.com/google/sentencepiece
- SamLynnEvans/Transformer https://github.com/SamLynnEvans/Transformer
- Huffon/pytorch-transformer-kor-eng https://github.com/Huffon/pytorch-transformer-kor-eng
- CLIP: Connecting text and images https://openai.com/index/clip/
- Transformer Explainer https://poloclub.github.io/transformer-explainer/
- OPUS corpora https://opus.nlpl.eu

THANK YOU!