



Transformer(Attention Is All You Need)

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c817c7a8-e9bf-47e6-ba1c-8fa91a8213d0/Transformer\(Attention Is All You Need\).pdf](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/c817c7a8-e9bf-47e6-ba1c-8fa91a8213d0/Transformer(Attention%20Is%20All%20You%20Need).pdf)

Code:

[Transformer Code1](#)

[Transformer Code2](#)

[Transformer Code3](#)

[Transformer 이해하기](#)

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5e89e9c0-3481-4924-999-de642857fb83/Transformer.py>

[노션](#)

1. 서론

- 기존의 모델은 인코더와 디코더를 포함하는 구조
- 인코더 디코더 기반으로 RNN이나 CNN을 사용

2. 관련 연구

▼ Sequential 문제를 풀기 위한, 이전 연구

Recurrent(순환) 구조는 Input과 Output Sequence를 활용

$h_t : t - 1$ 를 input과 h_{t-1} 를 통해 생성

→ 이러한 구조로 인해 일괄처리(병렬화)가 제한됨

(\because 순차적으로 t 이전의 Output이 다 계산되어야 최종 Output이 생성)

▼ Sequential Computation 문제를 풀기 위한, 이전 연구

▼ CNN 구조를 통해 병렬화

1. 인코더와 디코더를 연결하기 위한 추가 연산 필요
2. 원거리 Position간의 Dependencies(종속성)를 학습하기 어려움

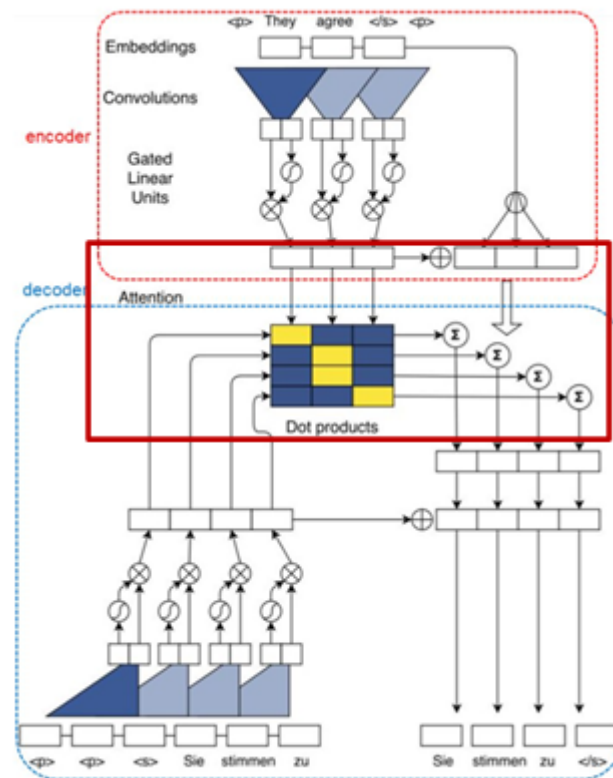


그림 1. ConvS2S 구조. CNN을 활용한 병렬화 방안

빨간 박스: 인코더와 디코더를 연결하기 위한 추가 연산 필요

Model Architecture

- Encoder-Decoder 구조를 가짐

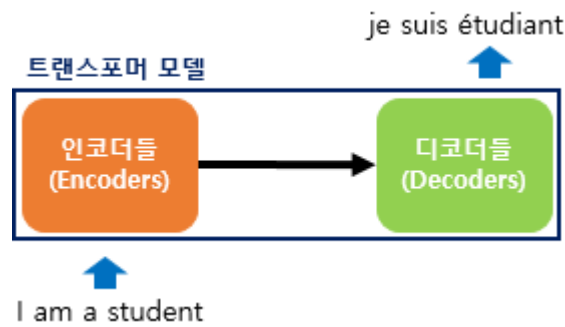


그림 1. 트랜스포머 구조

▼ 전체적인 구조

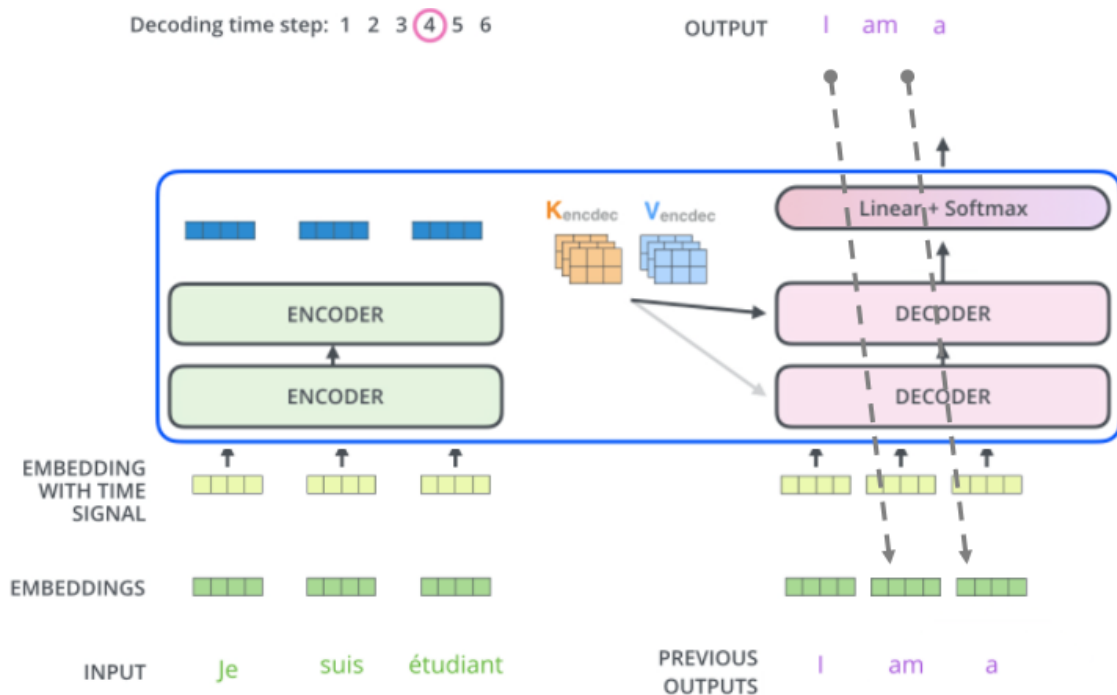


그림 2. Encoder-Decoder 구조

- 입력(x_1, \dots, x_n)은 Encoder를 통해 $z = (z_1, \dots, z_n)$ 로 표현 및 매핑됨
- Encoder로 표현한 z 를 활용하여, 한 Element씩 Output Sequence(y_1, \dots, y_m)가 생성



Auto-Regressive: 생성된 Symbol은 다음 생성 과정에서 추가 입력으로 사용

▼ Encoder 구조

- N=6의 동일한 Layer Stack으로 구성
- ▼ 각 Layer는 2개의 Sub-Layer로 구성
 1. Multi-Head Self-Attention 메커니즘
 - ▼ 2. Position-wise Fully Connected Feed-Forward Network
 - 1x1 Conv Layer가 2개 이어진 것과 같음
 - Position별로 동일한 Fully Connected Feed-Forward Network가 적용 (Dim=2048)
- 각 Sub-Layer는 Residual Connection 및 Layer Normalization 적용

$$LayerNorm(x + SubLayer(x))$$

- **Residual Connection** 적용을 용이하게 하기 위해

Sub-Layer, Embedding, Output Dimension을 512로 통일



Residual Connection 적용을 위해선

Input과 연결된 Output의 Dimension이 동일해야 함

▼ Attention 메커니즘

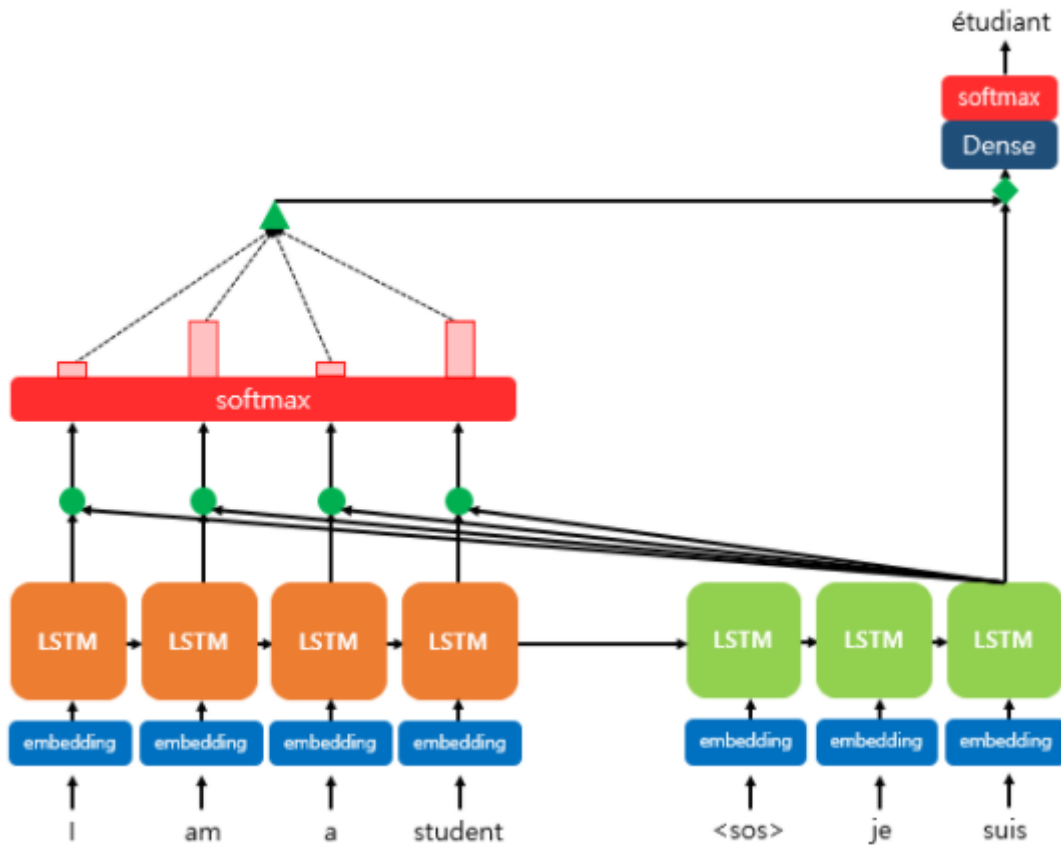


그림 2. Attention 구조

- input이나 Output Sequence에 관계없이 Dependencies(종속성)를 학습할 수 있음

Self-Attention 메커니즘

- 단일 Sequence 안에서 Position들을 연결
- 독해, 요약, Sentence Representation에서 효과적

요약

- 어텐션 메커니즘만을 사용하는 모델 제안

출처:

[\[\[논문요약\] Transformer 등장 - Attention Is All You Need\(2017\) ①\]](#)

[\[트랜스포머\(Transformer\) 간단히 이해하기 \(1\)\]](#)

1 Transformer Code 1

2 Transformer Code 2

3 Transformer Code 3