Attention is all you need

👨‍💻 저자

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin (Google brain)

✍️ 링크

- https://arxiv.org/pdf/1706.03762

1. Introduction

- **기존 모델의 한계:** RNN 계열 모델은 데이터를 순차적으로 처리해야 하므로 병렬 연산이 불가능함(h_t를 구하기 위해 h_(t-1)이 필수적). 이는 대규모 데이터 학습 시 시간적 비효율성을 초래함.
- **어텐션의 역할:** 거리와 상관없이 단어 간의 관계를 파악하는 어텐션은 유용하지만, 기존에는 RNN을 보조하는 용도로만 쓰였음.
- **트랜스포머:** "Attention Is All You Need"라는 제목처럼, RNN을 완전히 제거하고 오직 **Attention만으로** 모델을 구축하여 병렬 처리를 극대화하고 성능을 높임.

2. Background

- **Computational Efficiency:** 기존 CNN 기반 모델(ConvS2S, ByteNet)은 병렬 처리를 시도했으나, 임의의 두 지점 간의 신호를 연결하기 위한 연산량이 위치 간 거리에 의존함. 반면에 트랜스포머는 두 위치 사이의 거리에 관계없이 상수 횟수의 연산만으로 의존성을 모델링함.
- **Self-attention Relying:** RNN의 Sequence-aligned recurrence나 Convolution 구조를 완전히 배제하고 오직 **Self-attention**만으로 인풋/아웃풋 표현을 계산하는 독창적 아키텍처임.

3. Model Architecture

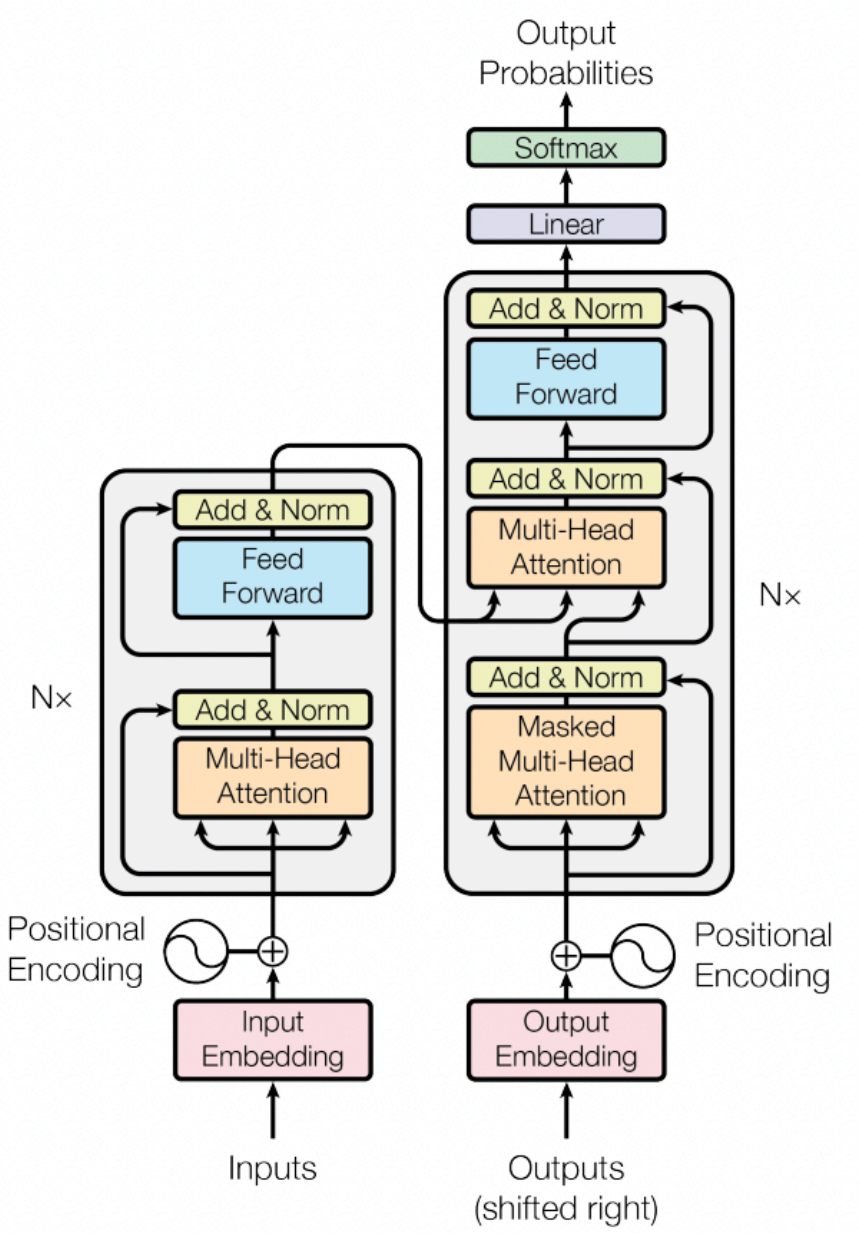


Figure 1: The Transformer - model architecture.

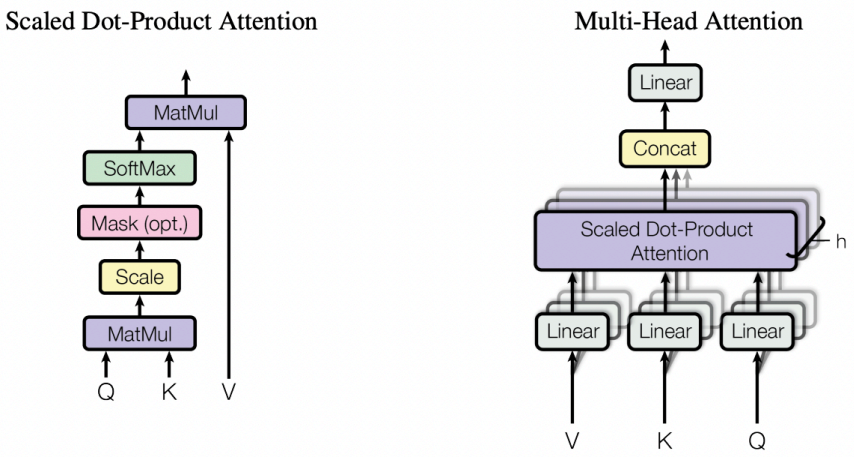
3.1) Encoder and Decoder Stacks

- 인코더 (Encoder)
  - 6개 층 적층: 동일한 구조의 레이어 6개를 쌓아 성능을 극대화함.
  - 두 개의 서브 레이어: Multi-Head Self-Attention (문장 내 단어 간 관계를 한꺼번에 파악). Position-wise FFN (각 단어의 의미를 비선형적으로 변환).
  - 잔차 연결 & 정규화: LayerNorm(x + Sublayer(x)) 구조를 사용해 학습 안정성을 확보하고 깊은 층에서도 정보 손실을 방지함.
- 디코더 (Decoder)

- 동일하게 6개의 층으로 구성.
- 세 개의 서브 레이어: 인코더 구조에 Encoder-Decoder Attention 층이 추가됨 (인코더의 정보를 가져오는 역할).
- 마스킹(Masking): 셀프 어텐션 시 현재 시점 이후의 단어를 보지 못하게 가려줌으로써, 미래 정보를 미리 알고 예측하는 오류를 방지.
- residual connections + layer normalization 이용.

3.2) Attention

어텐션 함수는 쿼리(Query)와 키(Key)-값(Value) 쌍의 집합을 출력으로 매핑하는 것.출력은 값(Value)들의 가중 합으로 계산되며, 각 값에 할당된 가중치는 쿼리와 해당 키의 호환성 함수에 의해 계산됨.



- **(3.2.1) Scaled Dot-Product Attention**
  - **계산 방식:** 쿼리(Q)와 키(K)를 점곱(Dot-product)한 뒤, 값(V)을 곱함.
  - **스케일링:** Gradient vanishing를 방지하기 위해 상숫값으로 나눠줌.
  - $Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$
  - 단어와 단어사이의 상관관계를 구하기 위해서, 같은 입력 행렬을 분리된 네트워크에 넣어서 Q, K행렬을 구한 뒤 곱하고 scale과 softmax를 적용하여 self-attention 값을 나타내는 행렬을 구하고 V행렬과 곱하여 입력+위치+어텐션 임베딩 행렬을 만듦.
  - 디코더의 masked multi-head attention에서 mask가 쓰이며, 디코더는 미래 즉 이후 부분의 단어들에 대해 어텐션 연산을 참조할 필요가 없으므로 mask 연산을

함. n=6으로 해당 부분이 반복될 때에도 계속 첫번째 multi-head attention은 mask가 쓰임 (feed forward에서 mask가 풀리기 때문).

- **(3.2.2) Multi-Head Attention**
  - **병렬 처리:** 하나의 큰 어텐션을 하는 대신, 작게 쪼갠 h=8개의 어텐션을 병렬로 수행. (d=512일때, 512*1이 아닌 64*8의 연산 수행)
  - **다양한 관점:** 모델이 문장 내의 여러 위치와 다양한 문맥적 의미(공간)를 동시에 학습할 수 있게 함.
- **(3.2.3) Applications of Attention in our Model**
  - **Encoder Self-Attention:** 입력 문장 내 관계 파악.
  - **Decoder Self-Attention:** 이미 생성된 단어 간 관계 파악.
  - **Encoder-Decoder Attention:** 디코더가 인코더의 출력값(입력 문장 정보)을 참조.

3.3) Position-wise Feed-Forward Networks

- $FFN(x) = \max(0, xW_1 + b_1)W_2 + b_2$
- **비선형성 부여:** 단순한 어텐션 연산만으로는 부족한 복잡한 데이터 패턴을 ReLU와 두 층의 신경망을 통해 학습함.

3.4) Embeddings and Softmax

- **차원 변환 (d = 512):** 텍스트 토큰을 모델이 처리할 수 있는 512차원의 연속적인 벡터 공간으로 매핑함.
- **가중치 공유 (Weight Sharing):** 입력 임베딩, 출력 임베딩, 그리고 최종 선형 변환 층에서 동일한 가중치 행렬을 사용함. 이는 모델의 파라미터 수를 줄이고 학습 효율을 높임.
- **스케일링 :** 임베딩 벡터에 이 값을 곱해줌으로써, 이후에 더해질 positional encoding과의 값 범위를 맞추고 학습을 안정화.

3.5) Positional Encoding

트랜스포머는 RNN처럼 단어를 순서대로 읽지 않고 한꺼번에 처리하기 때문에, "I"가 첫 번째고 "school"이 네 번째라는 위치 정보를 수동으로 넣어줘야 함.

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

- **순서 부여:** 시퀀스 데이터의 핵심인 '순서' 정보를 수동으로 결합함.
- **차원 유지:** 임베딩과 더하기 연산을 수행하여 최종 입력 차원을 **512**로 유지함.
- **유연성:** 학습 파라미터가 아닌 고정 함수를 써서 문장 길이에 대한 제약을 극복함.

---

4. Why Self-Attention

- **계산 효율성 (Complexity):** 보통 문장 길이(n)보다 차원 수(d=512)가 크기 때문에, O(n^2*d)인 셀프 어텐션이 $O(n*d^2)$인 RNN보다 연산 효율이 좋음.
- **병렬 처리 (Parallelism):** RNN은 이전 단어 계산을 기다려야 하지만, 셀프 어텐션은 모든 단어 관계를 한 번에 계산하므로 병렬화에 최적화됨.
- **장거리 의존성 (Path Length):** 문장 양 끝에 있는 단어라도 단 한 번의 단계(O(1))로 연결됨. 이는 정보 손실 없이 긴 문장을 처리하는 데 결정적임.
- **시각화 및 해석 (Interpretability):** 어텐션 가중치를 통해 모델이 어떤 단어에 집중하는지 시각적으로 확인 가능함.

---

5. Training

5.1) Training Data and Batching

- WMT 2014 영어-독일어 데이터셋(450만개 문장), 영어-프랑스어 데이터셋(3600만개)
- 토큰화에는 BPE 및 Word-piece 방식 사용.

5.2) Hardware and Schedule

- 8개 NVDIA P100 GPU 사용.
- base model 10만 스텝(약 12h), big model 30만 스텝(3.5 day) 동안 학습.

5.3) Optimizer

- Adam 사용
- 학습률 고정하지 않고 변화하는 방식.

5.4) Regularizaiton

- Dropout: 각 서브 레이어의 출력(더하기 및 정규화 전)과 임베딩 합계에 드롭아웃(p=0.1)을 적용.
- Label Smoothing: 모델이 정답에 너무 확신하지 않도록 (epsiolon=0.1)의 라벨 스무딩을 적용. 이는 Perplexity는 낮추지만, 실제 정확도와 BLEU 점수를 향상.

---

6. Results

6.1) Machine Translation

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

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

- WMT 2014 영-독 번역에서 **Transformer (big)** 모델은 기존의 최고 기록(앙상블 포함)을 2.0 BLEU 이상 경신하며 **28.4 BLEU**로 새로운 SOTA를 달성. 영-프 번역에서도 기존 최고 모델 학습 비용의 1/4 미만을 사용하고도 **41.0 BLEU 기록.**

6.2) Model Variations

Table 3: Variations on the Transformer architecture. Unlisted values are identical to those of the base model. All metrics are on the English-to-German translation development set, newstest2013. Listed perplexities are per-wordpiece, according to our byte-pair encoding, and should not be compared to per-word perplexities.

	N	d_{model}	d_{ff}	h	d_k	d_v	P_{drop}	ϵ_{ls}	train steps	PPL (dev)	BLEU (dev)	params $\times 10^6$
base	6	512	2048	8	64	64	0.1	0.1	100K	4.92	25.8	65
(A)				1	512	512				5.29	24.9	
				4	128	128				5.00	25.5	
				16	32	32				4.91	25.8	
				32	16	16				5.01	25.4	
(B)					16					5.16	25.1	58
					32					5.01	25.4	60
(C)	2									6.11	23.7	36
	4									5.19	25.3	50
	8									4.88	25.5	80
		256			32	32				5.75	24.5	28
		1024			128	128				4.66	26.0	168
			1024							5.12	25.4	53
			4096							4.75	26.2	90
(D)							0.0			5.77	24.6	
							0.2			4.95	25.5	
								0.0		4.67	25.3	
								0.2		5.47	25.7	
(E)	positional embedding instead of sinusoids									4.92	25.7	
big	6	1024	4096	16			0.3		300K	**4.33**	**26.4**	213

- 트랜스포머의 각 구성 요소가 성능에 미치는 영향을 평가하기 위해, 기본 모델(Base model)을 바탕으로 하이퍼파라미터를 변화시키며 영어-독일어 번역 성능(BLEU)을 측정.
  - **(A) Multi-head Attention의 크기 조정:** 연산량은 일정하게 유지하면서 헤드 수(h)와 키/값 차원(d_k, d_v)을 조절. 헤드가 너무 적거나(1개) 너무 많아도 성능이 떨어졌으며, h=8일 때 가장 좋은 결과를 보였음.
  - **(B) Attention Key 크기(d_k)의 영향:** d_k를 줄였을 때 성능이 크게 저하됨. 이는 두 단어 사이의 유사성(Compatibility)을 판단하는 작업이 단순하지 않으며, 차원이 어느 정도 확보되어야 함을 뜻
  - **(C) & (D) 모델 크기 및 규제:** 모델이 깊고 넓을수록(d_{model}, d_{ff} 증가), 그리고 레이어 수가 많을수록 성능이 향상. 또한 드롭아웃(P_{drop})이 과적합 방지에 결정적인 역할을 함을 확인함.
  - **(E) 위치 인코딩 방식:** 사인파를 이용한 고정 방식(Sinusoidal)과 학습 가능한 임베딩 방식(Learned)을 비교했으나 성능 차이가 거의 없었음.

6.3) English Constituency Parsing

Table 4: The Transformer generalizes well to English constituency parsing (Results are on Section 23 of WSJ)

Parser	Training	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37]	WSJ only, discriminative	88.3
Petrov et al. (2006) [29]	WSJ only, discriminative	90.4
Zhu et al. (2013) [40]	WSJ only, discriminative	90.4
Dyer et al. (2016) [8]	WSJ only, discriminative	91.7
Transformer (4 layers)	WSJ only, discriminative	91.3
Zhu et al. (2013) [40]	semi-supervised	91.3
Huang & Harper (2009) [14]	semi-supervised	91.3
McClosky et al. (2006) [26]	semi-supervised	92.1
Vinyals & Kaiser el al. (2014) [37]	semi-supervised	92.1
Transformer (4 layers)	semi-supervised	92.7
Luong et al. (2015) [23]	multi-task	93.0
Dyer et al. (2016) [8]	generative	93.3

- **영어 구문 분석 (Parsing):** 번역 외의 일반화 능력을 측정하기 위해 수행. 트랜스포머는 작업별 특화 튜닝 없이도 매우 뛰어난 성과를 냈으며, 데이터가 적은 상황에서도 기존 RNN 기반 모델들보다 훨씬 우수한 일반화 성능을 보여줌.

---

7. Conclusion

- **패러다임의 전환:** "Attention Is All You Need"라는 제목처럼, 복잡한 RNN 없이 Attention 만으로 충분하다는 것을 보여줌.
- **압도적인 효율과 성능:** 학습 속도는 혁신적으로 높이면서 성능(BLEU)까지 동시에 잡음.
- **확장 가능성:** 텍스트를 넘어 이미지, 오디오 등 **멀티모달(Multi-modal)**로 나아갈 발판을 마련함. (실제로 이후 Vision Transformer 등으로 이어짐)