Attention Is All You Need

 “순환 신경망(RNN) 대신 **Attention** 구조만을 활용해 보자.”

1. Introduction

- 순환 신경망과 어텐션 메커니즘은 순차적 모델링에서 중요한 역할을 수행
  - 그러나 순환 신경망은 병렬화가 어렵고 메모리 소모가 크다는 단점이 존재
- 이러한 한계를 극복하기 위해 어텐션 메커니즘만을 활용하는 `Transformer` 라는 새로운 모델이 제안되었음
  - `Transformer` 는 병렬화가 용이하며, 더 빠르게 훈련되면서도 번역 품질을 향상시킬 수 있는 잠재력을 보임

2. Background

Extended Neural GPU, ByteNet 및 ConvS2S

- 시퀀셜 계산을 줄이는 것이 목표
- 컨볼루션 신경망(CNN)을 기본 구성 요소로 사용
- 모든 입력 및 출력 위치에 대해 병렬로 숨겨진 표현을 계산
- 거리가 증가하면 신호를 관련시키는 데 필요한 연산 수가 증가

Self-Attention

- 단일 시퀀스의 다양한 위치를 관련시켜 시퀀스의 표현을 계산하는 어텐션 메커니즘
- 읽기 이해, 추상적 요약, 텍스트 연역, 작업 독립적인 문장 표현 학습 등 다양한 작업에서 성공적으로 활용됨

End-to-end 메모리 네트워크

- 순차적으로 정렬된 순환 대신 반복적인 어텐션 메커니즘을 기반으로 함

- 간단한 언어 질문 응답 및 언어 모델링 작업에 특화

`Transformer`는 시퀀스 정렬된 RNN이나 컨볼루션을 사용하지 않고 입력과 출력의 표현을 계산하기 위해 완전히 **셀프 어텐션**에 의존하는 최초의 변환 모델임

3. Model Architecture

- 대부분의 순서 기반 모델은 대부분 Encoder-Decoder 구조를 갖추고 있음
  - 인코더: 기호 표현의 입력 시퀀스 $(x_1, \cdots, x_n)$를 연속적인 표현 시퀀스 $z = (z_1, \cdots, z_n)$로 매핑
  - 디코더: z가 주어지면 하나의 기호 출력 시퀀스 $(y_1, \cdots, y_m)$를 생성
- 각 단계에서 모델은 자기 회귀적이며, 다음 값을 생성할 때 이전에 생성된 기호를 추가 입력으로 사용함
- Transformer는 이러한 전반적인 구조를 따름
  - 인코더와 디코더 모두에 대해 쌓인 self-attention과 포인트별로 완전히 연결된 레이어를 사용

3-1. Encoder and Decoder Stacks

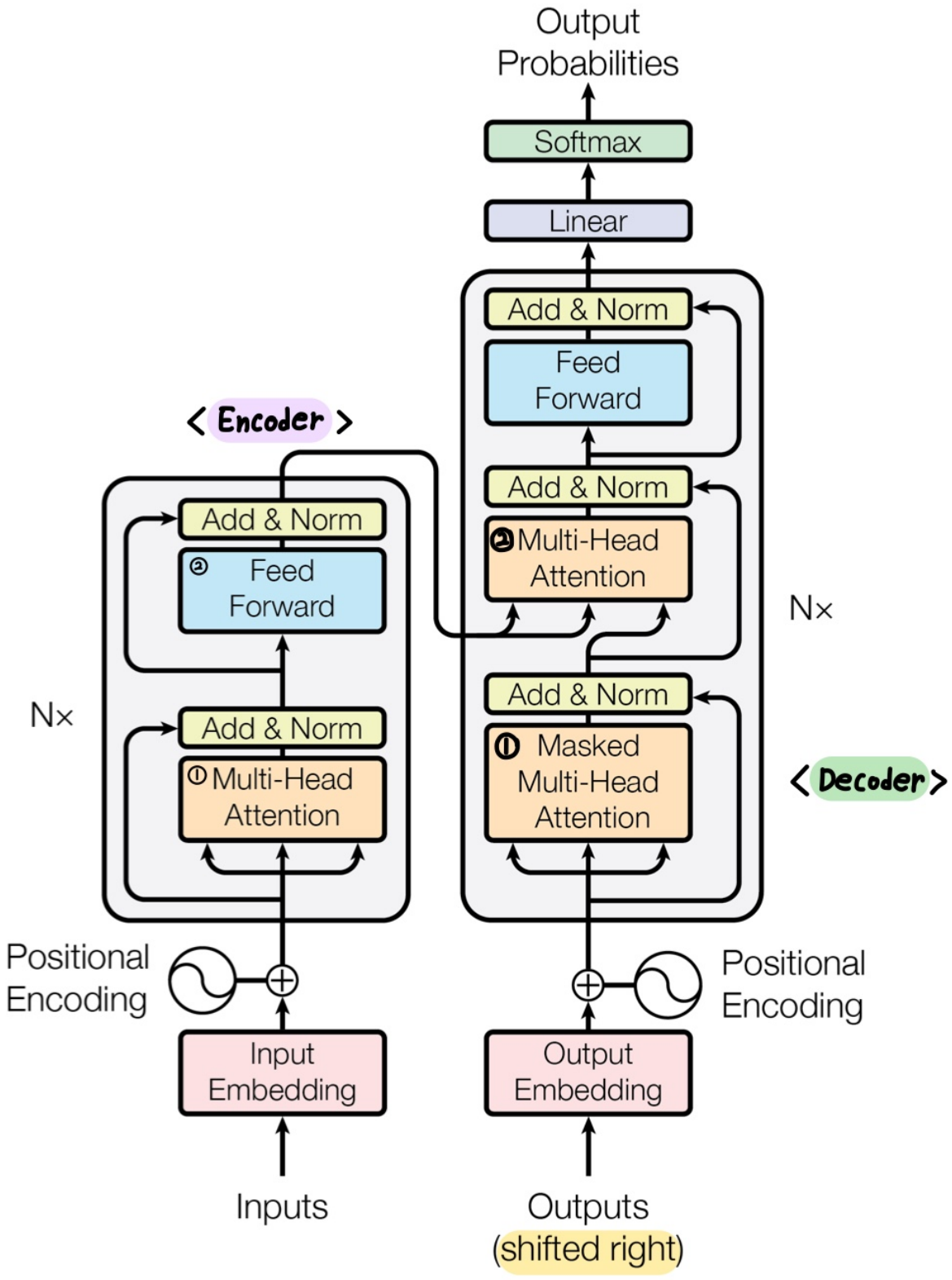


Figure 1: The Transformer - model architecture.

- **인코더(Encoder)**
  - 6개의 동일한 레이어로 구성되어 있음

    - 각 레이어에는 멀티 헤드 셀프 어텐션과 위치별 Fully-Connected feed-forward 네트워크가 존재
    - 잔차 연결과 레이어 정규화가 적용되어 있음
    - 출력 차원: 512
- **디코더(Decoder)**
    - 6개의 동일한 레이어로 구성되어 있음
    - 각 레이어는 인코더와 동일하게 구성됨
        - 추가로 인코더 stack의 출력을 대상으로 하는 multi-head 어텐션을 수행하는 세 번째 서브 레이어가 존재
    - 위치에 대한 마스킹이 적용되어 있음
        - 각 위치의 예측은 해당 위치 이전의 출력에만 의존

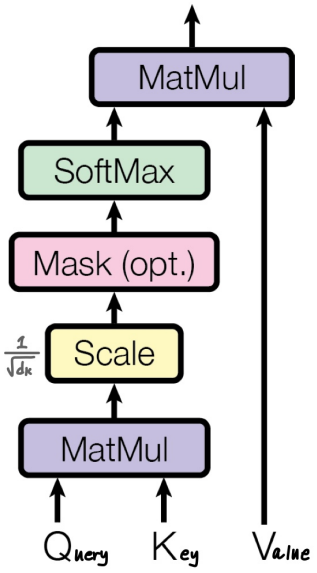
3-2. Attention

- 어텐션 함수
    - 쿼리와 키-값 쌍의 집합을 출력으로 매핑하는 함수
        - 쿼리, 키, 값 및 출력은 모두 벡터
    - 출력은 각 값에 할당된 가중합으로 계산됨
        - 각 값에 할당된 가중치는 해당 키와의 호환성 함수에 의해 계산됨

3-2-1. Scaled Dot-Product Attention

- 쿼리(Q)와 키(K)의 dot product를 스케일링한 후 softmax 함수를 적용하여 값에 가중치를 부여

Scaled Dot-Product Attention

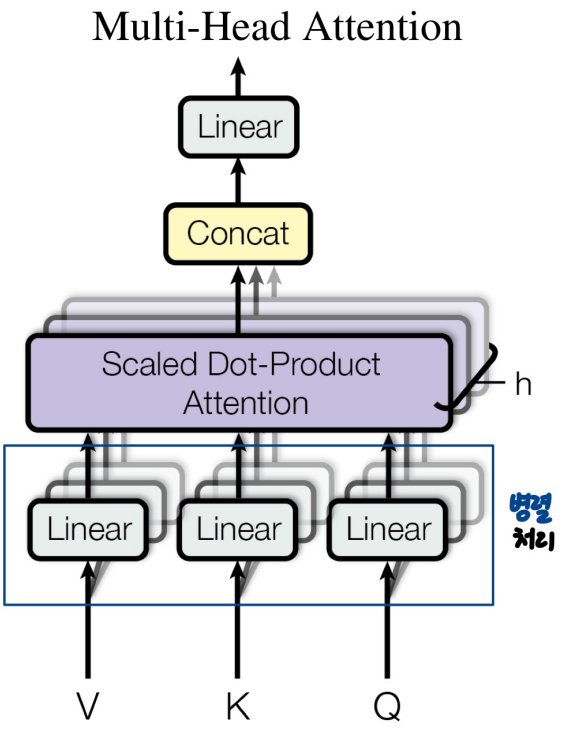


$$\text{Attention}(\underset{\text{query}}{Q^{\in d_k}}, \underset{\text{key}}{K^{\in d_k}}, \underset{\text{value}}{V^{\in d_v}}) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

- 흔히 사용되는 어텐션 함수 중에는 addition attention과 dot-product attention이 있음
  - addition attention은 단일 hidden 레이어를 사용하여 호환성을 계산
  - dot-product attention은 행렬곱 코드를 사용하여 더 빠르고 공간 효율적인 계산을 가능하도록 함
- 대부분의 경우에는 두 메커니즘이 유사하게 동작함
  - 그러나 큰 d_k값에 대해서는 dot-product attention이 작은 기울기 영역으로 소프트맥스 함수를 밀어 넣어 문제가 발생할 수 있음
    - 이를 해결하기 위해 dot-product를 $\frac{1}{\sqrt{d_k}}$로 스케일링

3-2-2. Multi-Head Attention

- 단일 attention 함수를 적용하는 대신 linear projection을 활용하여 병렬 처리를 수행



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

linear projection

병렬 처리

    - 투영된 버전의 쿼리, 키 및 값에 대해 병렬로 어텐션 함수를 수행하고, 출력 값을 생성
    - 출력 값은 연결되어 다시 투영, 최종 값을 생성
- multi-head attention을 사용하면 모델이 서로 다른 위치에서 서로 다른 표현 하위 공간의 정보에 공동으로 주의를 기울일 수 있음
- h = 8 병렬 attention layer 또는 head 활용
    - 각각에 대해 $d_k = d_v = d_{model}/h = 64$ 활용
    - 계산 비용은 단일 attention head를 활용할 때와 비슷

3-2-3. Applications of Attention in our Model

- Transformer는 multi-head attention을 세 가지 방식으로 사용
    - 인코더-디코더 어텐션에서는 디코더의 각 위치가 입력 시퀀스의 모든 위치에 주의를 기울일 수 있음

    - 인코더의 셀프-어텐션 레이어는 각 위치가 이전 레이어의 모든 위치에 주의를 기울일 수 있음
    - 디코더의 셀프-어텐션 레이어는 각 위치가 이전 디코더의 모든 위치에 주의를 기울일 수 있음
        - 이를 통해 왼쪽 방향의 정보 흐름을 방지하여 자기 회귀 속성을 보존

3-3. Position-wise Feed-Forward Networks

- attention sub-layer와 더불어 각 encoder/decoder 층은 fully connected feed-forward 네트워크를 가지고 있음
    - ReLU 활성화 함수를 거치는 두 개의 선형 변환으로 구성

$$\mathrm{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2$$

- 입력/출력 차원: $d_{model} = 512$
- 내부 레이어: $d_{ff} = 2048$

3-4. Embeddings and Softmax

- 입력 토큰과 출력 토큰을 d_{model} 차원의 벡터로 변환하기 위해 학습된 임베딩을 사용
- 또한 디코더 출력을 예측된 다음 토큰 확률로 변환하기 위해 선형 변환과 소프트맥스 함수를 사용
- 우리 모델에서는 두 임베딩 레이어와 소프트맥스 이전의 선형 변환 사이에 동일한 가중치 행렬을 공유
    - 임베딩 레이어에서는 이러한 가중치를 $\sqrt{d_{model}}$로 곱함

3-5. Positional Encoding

- Transformer 모델은 순환 및 컨볼루션 없이 시퀀스의 순서를 활용하기 위해 **위치 인코딩**을 사용
- 위치 인코딩은 입력 임베딩에 추가되며, 각 차원은 사인 및 코사인 함수를 사용하여 다른 주파수의 파형으로 표현됨

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

    - 이를 통해 모델은 상대적 위치에 대한 주의를 쉽게 학습할 수 있음
- 실험 결과, 학습된 위치 임베딩과 사인 파형 버전의 성능은 거의 동일하였음
    - 그러나 sine 파형 버전이 더 긴 시퀀스 길이에 대한 추정을 가능하게 함을 확인

4. Why Self-Attention

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

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

- self-attention layer와 일반적으로 사용되는 순환 및 컨볼루션 레이어를 여러 측면에서 비교
    - 각 레이어 유형의 총 계산 복잡도를 고려
    - 병렬화 가능 계산량 평가
        - 순차적으로 수행되는 최소 연산 횟수로 측정
    - 네트워크 내의 장거리 종속성에 대한 경로 길이를 비교
- self-attention layer는 모든 위치를 일정한 수의 순차적 연산으로 연결하기 때문에 계산 복잡성 면에서 효율적임
    - 이와 달리, 순환 레이어는 $O(n)$의 순차적 연산 필요
- 또한, self-attention은 상대적으로 병렬화 하기 쉬움
    - 한편, 컨볼루션 레이어는 커널 폭에 따라 최대 경로 길이가 달라지며, 복잡도도 더 높음
    - 인접한 커널을 사용하는 경우 $O(n/k)$ 개의 컨볼루션 레이어 스택이 필요하며, 이는 네트워크 내의 최대 경로 길이를 증가시킴
- 마지막으로, self-attention layer은 해석 가능한 모델을 제공할 수 있음

- 개별 attention head가 다른 작업을 수행하고 문장의 구문 및 의미 구조와 관련된 특징을 학습하는 것으로 나타남

5. Training

5-1. Training Data and Batching

- 약 450만 개의 문장 쌍으로 구성된 표준 WMT 2014 영어-독일어 데이터셋으로 훈련
  - 바이트 페어 인코딩을 사용하여 인코딩을 진행하였음
  - 약 37000개의 토큰으로 구성된 공유 소스-타겟 어휘를 사용
- 영어-프랑스어의 경우, 3600만 개의 문장으로 구성된 큰 WMT 2014 영어-프랑스어 데이터셋을 사용했음
  - 토큰은 32000개의 워드피스 어휘로 분할되었음
- 문장 쌍은 근사적인 시퀀스 길이에 따라 함께 배치되었음
- 각 훈련 배치는 약 25000개의 소스 토큰과 25000개의 타겟 토큰을 포함하는 문장 쌍 세트를 포함

5-2. Hardware and Schedule

- 8개의 NVIDIA P100 GPU를 장착한 한 대의 컴퓨터에서 모델을 훈련시켰음
- 논문 전체에 걸쳐 설명된 하이퍼파라미터를 사용한 기본 모델의 경우, 각 훈련 단계는 약 0.4초가 소요되었음
  - 기본 모델을 총 100,000 단계 또는 12시간 동안 훈련
- 큰 모델의 경우, 단계 시간은 1.0초였음
  - 큰 모델은 300,000 단계(3.5일) 동안 훈련

5-3. Optimizer

- `Adam` 활용
  - $\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 10^{-9}$
- 아래 formula에 따라 learning rate를 업데이트

$$lrate = d_{\text{model}}^{-0.5} \cdot \min(step\_num^{-0.5}, step\_num \cdot warmup\_steps^{-1.5})$$

- 처음: $warmup_steps = 4000$에서 시작
- 이후 계속 줄여나가는 방식

5-4. Regularization

Residual Dropout

- 각 sub-layer의 출력에 드롭아웃을 적용하여 sub-layer 입력에 추가되기 전에 정규화
- encoder 및 decoder stack의 embedding 및 위치 인코딩의 합에도 드롭아웃을 적용
- 기본 모델에서는 드롭아웃 비율을 $P_{drop} = 0.1$로 사용

Label Smoothing

- 훈련: $\epsilon_{ls} = 0.1$ 적용
  - 모델이 더 불확실해지도록 학습 → perplexity에 악영향
  - 그러나 정확도와 BLEU 점수를 향상시킴

▼ BLEU Score

- Bilingual Evaluation Understudy
- 기계 번역 결과와 사람이 직접 번역한 결과가 얼마나 유사한지 비교하여 번역에 대한 성능을 측정하는 방법
- n-gram에 기반하여 측정
  - 높을수록 성능이 더 좋음을 의미
- 언어에 구애받지 않고 사용할 수 있으며, 계산 속도가 빠르다는 장점이 있음

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 영어-독일어 번역 작업에서 큰 트랜스포머 모델은 이전 모델보다 2.0 BLEU 이상 높은 성능을 보였으며, 새로운 최고 수준의 BLEU 점수를 기록하였음
- 영어-프랑스어 번역 작업에서도 (우리의) 큰 모델은 이전의 단일 모델을 능가했으며, 훈련 비용은 이전 최고 수준 모델의 1/4 이하였음
- 훈련에는 8개의 P100 GPU를 사용하여 3.5일이 소요되었음
    - 모델 훈련에는 드롭아웃 및 라벨 스무딩과 같은 기술을 사용했으며, 이는 모델의 정확도와 BLEU 점수를 향상시켰음

6-2. Model Variations

- `Transformer`의 다양한 구성 요소의 중요성을 평가하기 위해, 베이스 모델을 다양한 방식으로 변형하고 영어-독일어 번역 개발 세트인 newstest2013에서 성능 변화를 측정하였음

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

- **(A)**
  - 어텐션 헤드 수와 어텐션 키 및 값 차원을 변경하여 계산 양을 일정하게 유지하였음
  - 단일 헤드 어텐션은 최적 설정보다 0.9 BLEU 낮으나, 너무 많은 헤드로 인해 품질이 떨어짐
- (B)
  - 키 크기를 줄이는 것이 모델 품질에 악영향을 미침을 확인

    → 호환성을 결정하는 것이 쉽지 않으며, 점곱 이상의 더 정교한 호환성 함수가 유리할 수 있다는 것을 시사
- (C), (D)
  - 예상대로 큰 모델이 더 나은 성능을 보이며, 드롭아웃은 과적합을 피하는 데 매우 도움이 됨
- (E)
  - 주기적 위치 인코딩을 학습된 위치 임베딩으로 대체하여 기본 모델과 거의 동일한 결과를 얻었음

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 구성 분석에 대한 실험에서 Transformer는 이전에 보고된 대부분의 모델을 능가하며, 작업별 튜닝이 없어도 놀랍도록 우수한 성능을 보임
- RNN seq-to-seq 모델과 달리 Transformer는 작은 데이터 환경에서도 뛰어난 성능을 보이며, WSJ 훈련 세트에서만 훈련할 때도 Berkeley-Parser보다 우수한 결과를 보임

7. Conclusion

- 해당 연구에서는 **Transformer**를 소개하였음
  - 이는 주로 인코더-디코더 아키텍처에서 사용되는 순환 레이어를 완전히 대체하고 멀티 헤드 셀프 어텐션을 사용한 첫 번째 시퀀스 변환 모델임
- 번역 작업의 경우, Transformer는 순환 또는 합성곱 레이어를 기반으로 하는 아키텍처보다 훨씬 빠르게 훈련될 수 있음
  - WMT 2014 영어-독일어 및 WMT 2014 영어-프랑스어 번역 작업에서 SOTA 달성
- 어텐션 기반 모델이 텍스트 이외의 입력 및 출력 모달리티를 다루는 문제에 Transformer를 확장하고, 이미지, 오디오 및 비디오와 같은 대규모 입력 및 출력을 효율적으로 처리하기 위해 로컬 및 제한된 어텐션 메커니즘을 연구할 계획
- 생성 과정을 덜 순차적으로 만드는 것 또한 연구 목표 중 하나임