Attention Is All You Need
Paper review

모바일시스템공학과
이승재

1. Introduction

RNN (1986)

- Recurrent Neural Network
- 고정된 길이의 문맥벡터 사용
- 마지막 은닉층의 값이 문맥벡터

LSTM (1997)

- Long Short-Term Memory
- RNN에서 장기적 기억을 더 잘하도록 변환된 신경망

Seq2Seq (2014)

- 시퀀스 입력을 통해 다른 시퀀스 출력을 얻도록 고안된 모델
- 기계번역에 주로 사용되는 모델

Attention (2015)

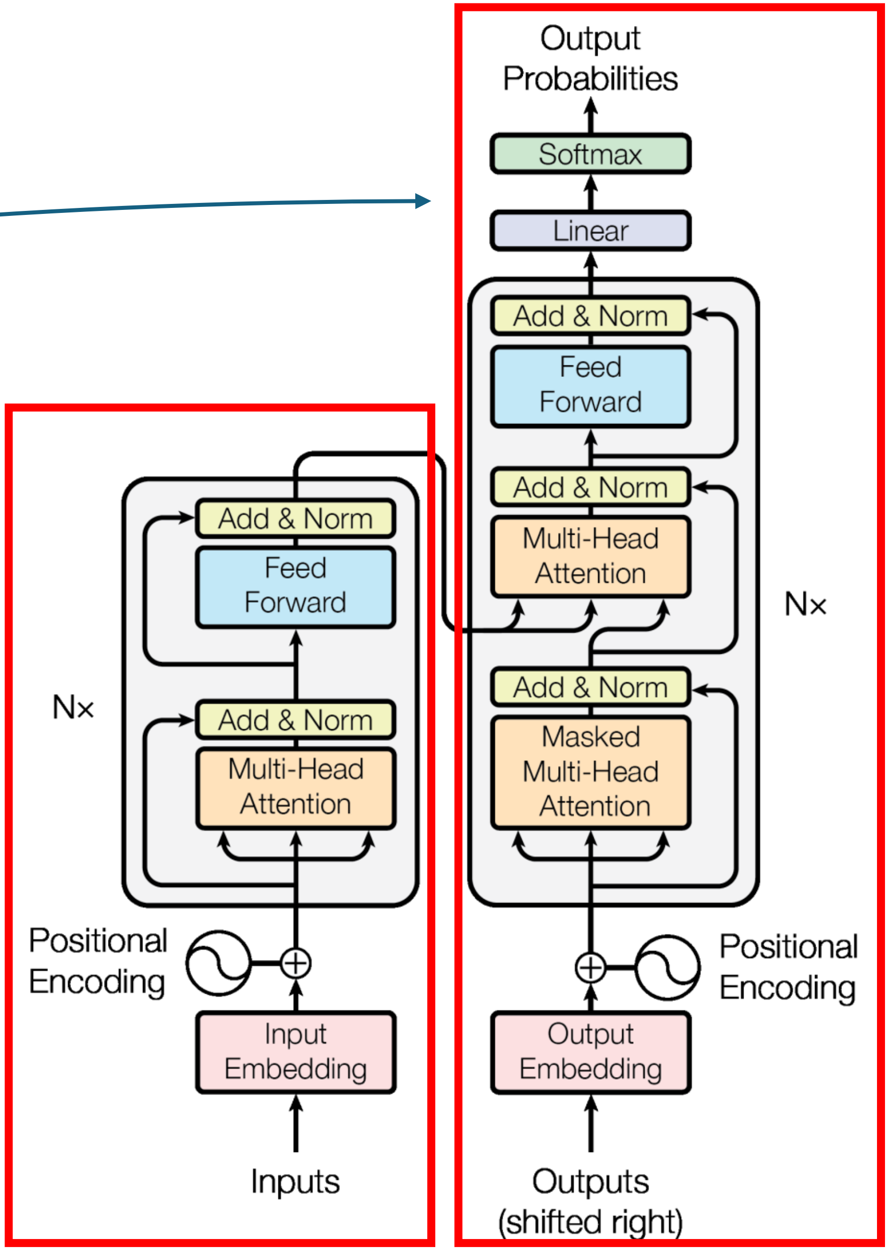
- 고정길이 문맥벡터를 사용하지 않아 Bottleneck 문제를 보완
- 전체 입력 시퀀스에서 '주의'를 기울여야하는 부분에 초점을 맞춰 학습

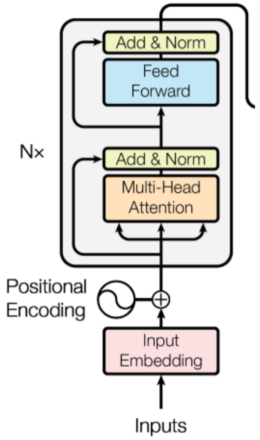
Transformer (2017)

- Attention만을 활용하여 구성한 Encoder-Decoder 모델
- 연산효율성과 기계 번역 품질에서 높은 성능을 낸 모델
- BERT, GPT의 기반 모델

2. Background – overall architecture

“The Transformer model consists of an **encoder-decoder architecture** designed for sequence-to-sequence tasks, such as machine translation.”





2. Background – partial architecture (Input Embedding)

"In the **Input Embedding** step of the Transformer, Input sentences are processed and converted into numerical vectors that the model can understand."

Tokenization **Embedding**

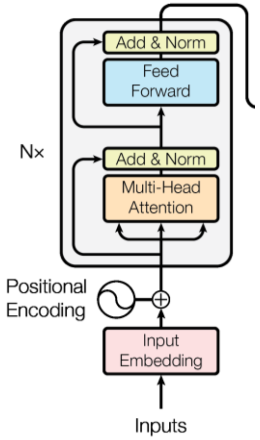
"I am a student" → [I, am, a, student] → $e_0,\ e_1,\ e_2,\ e_3$

② ③ ④ ⑤ $(e_i \in R^E)$

$d_{model} = 512$

②	V_{21}	V_{22}	…	V_{2511}	V_{2512}
③	V_{31}		…		
④	V_{41}		…		
⑤	V_{51}		…		

2. Background – partial architecture (Positional Encoding)



"**Positional Encoding** adds unique, continuous position-specific information to input embeddings using sinusoidal functions, enabling the Transformer to capture sequential relationships without recurrence."

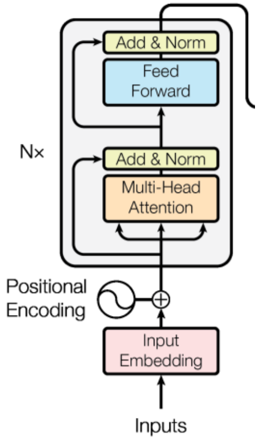
d_{model} = 512

PE(0)	V_{21}	V_{22}	…	V_{2511}	V_{2512}
PE(1)	V_{31}		…		
PE(2)	V_{41}		…		
PE(3)	V_{51}		…		

(pos, i)

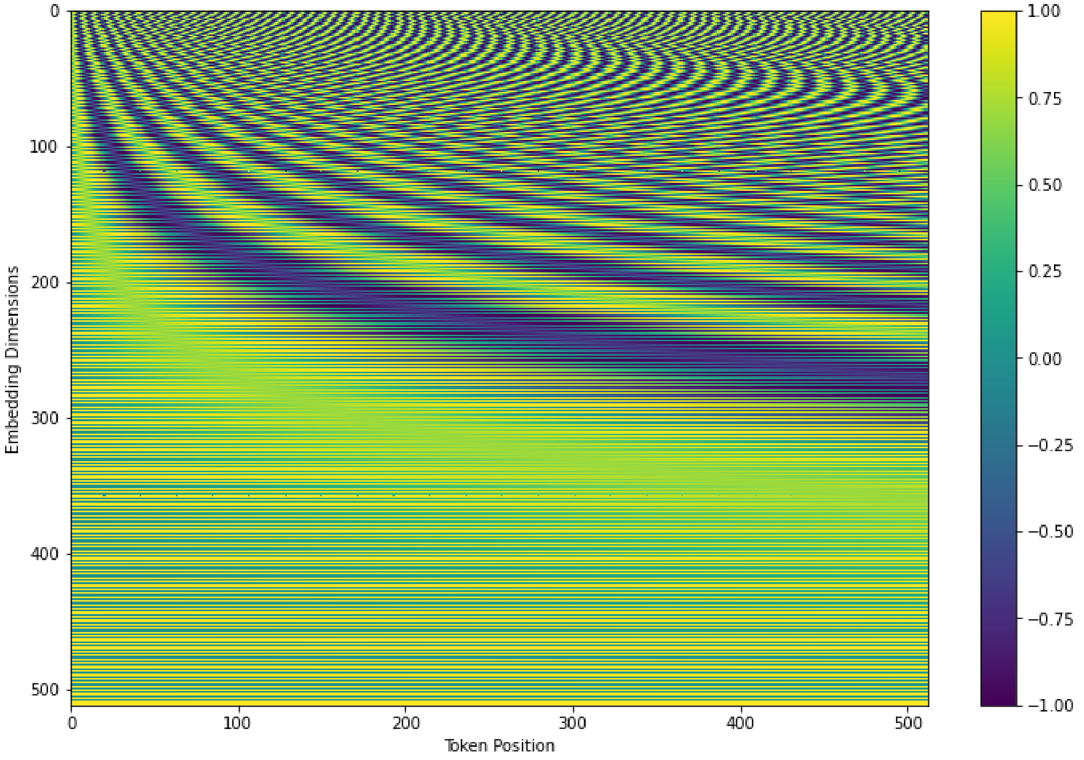
- Pos : Token location
- i : i-th value of the embedding vector

2. Background – partial architecture (Positional Encoding)



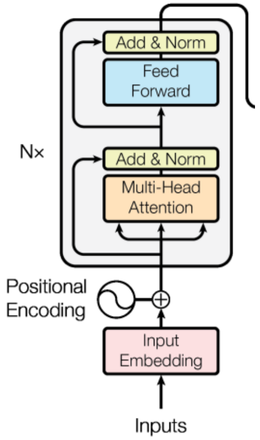
$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}})$$

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



"This graph visualizes how **Positional Encoding** adds positional information to the input data. Based on the following formula, lower dimensions have longer periods, which help in learning relationships between distant words, while higher dimensions have shorter periods, aiding in learning relationships between nearby words."

2. Background – partial architecture (Intrance of Self-Attention)



$$\begin{bmatrix} V_2 & & PE(0) \\ V_3 & \oplus & PE(1) \\ V_4 & & PE(2) \\ V_5 & & PE(3) \end{bmatrix}$$

X

$$X \cdot W_Q = Q$$

$$X \cdot W_k = K$$

$$X \cdot W_V = V$$

Dimensions of Q, K, V

$= d_{model}$ / num_heads

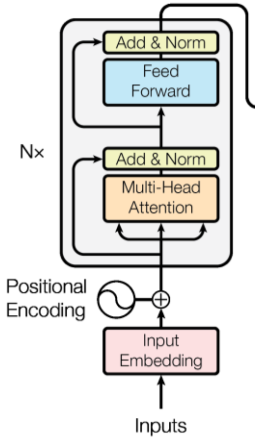
$= 512 / 6 = 64$

<4 x 512>

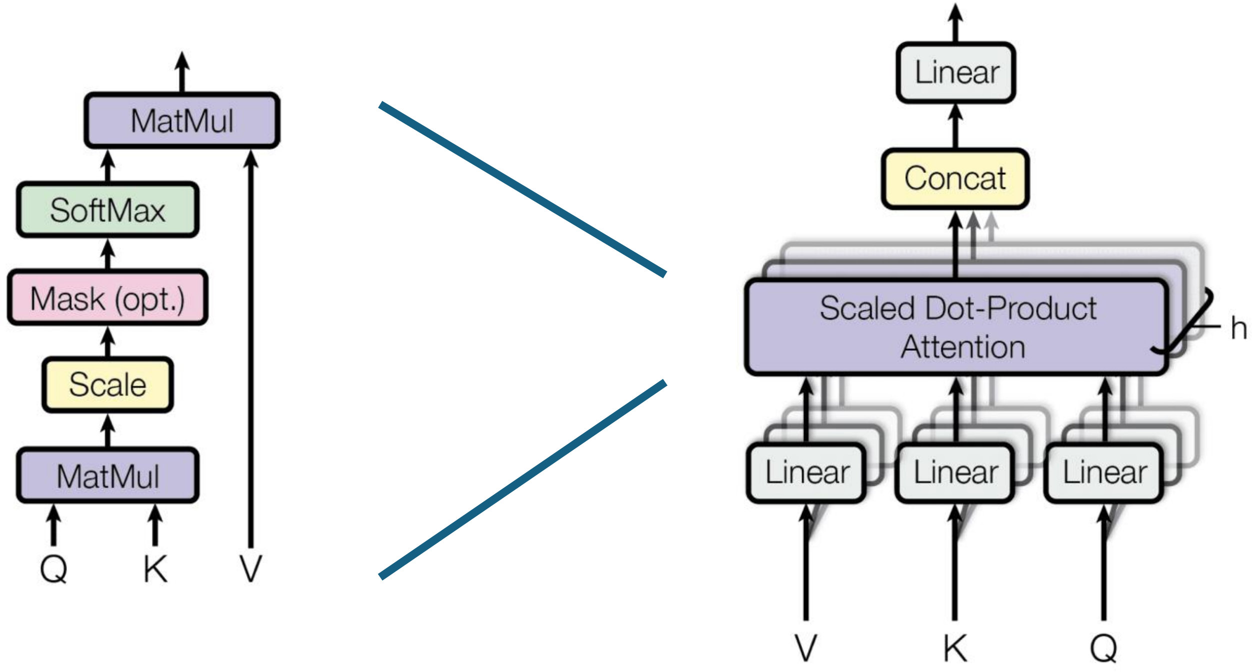
<4 x 512> · <512 x 64> = <4 x 64>

Q, K, V는 512x512이지만 multi-head로 분할하면 512x64가 됨

2. Background – partial architecture (Multi-Head Self Attention)

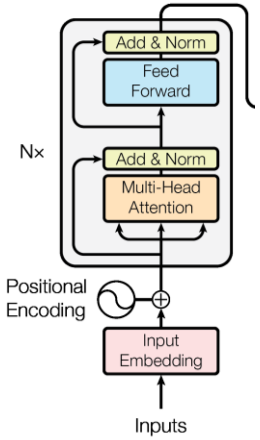


Scaled Dot-Product Attention



"**Multi-Head Self-Attention** splits the input into multiple heads, enabling the model to understand the relationships between words in the sequence effectively. **Scaled Dot-Product Attention** is applied independently to each head, and the results are concatenated and linearly transformed."

2. Background – partial architecture (Multi-Head Self Attention)



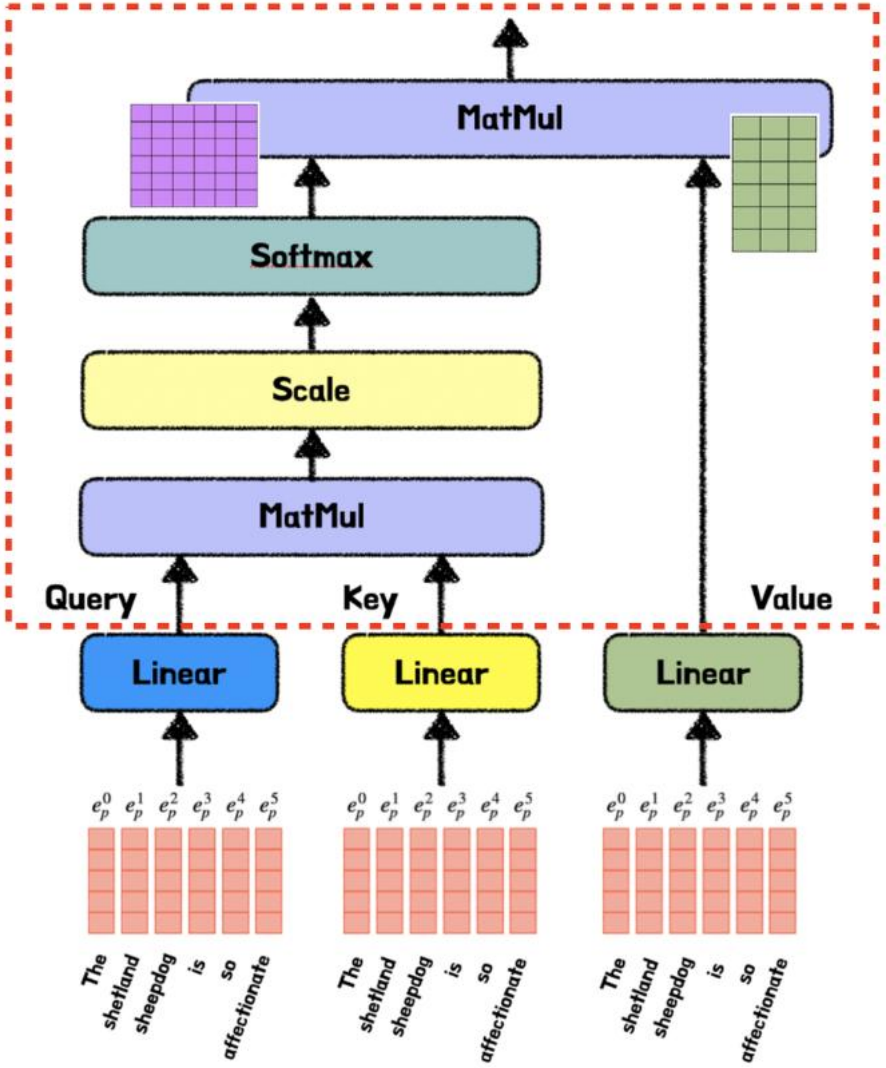
$$\mathrm{Attention}(Q,K,V) = \boxed{\mathrm{softmax}\left(\frac{\boxed{QK^T}}{\sqrt{d_k}}\right)}V$$

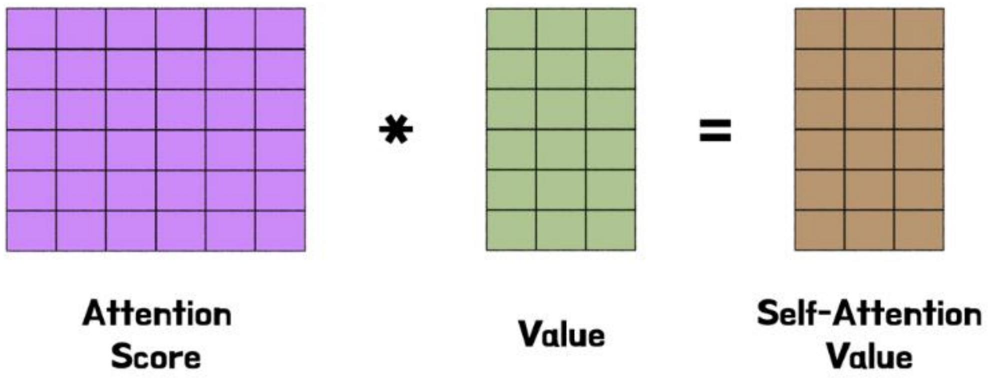
Attention score to be given to the relationship between the i-th token and the j-th token among the entire inner product of Q and K

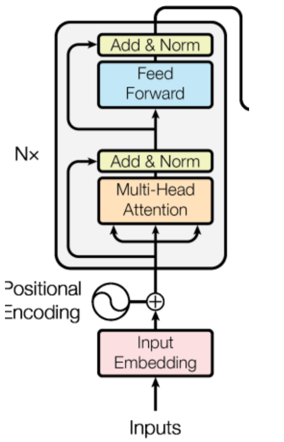
Weight indicating how important a word is to other words

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head_1},\ldots,\mathrm{head_h})W^O$$
$$\text{where } \mathrm{head_i} = \mathrm{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

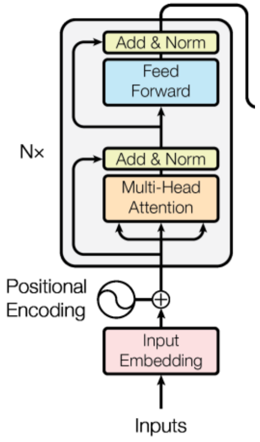
2. Background – partial architecture (Multi-Head Self Attention)



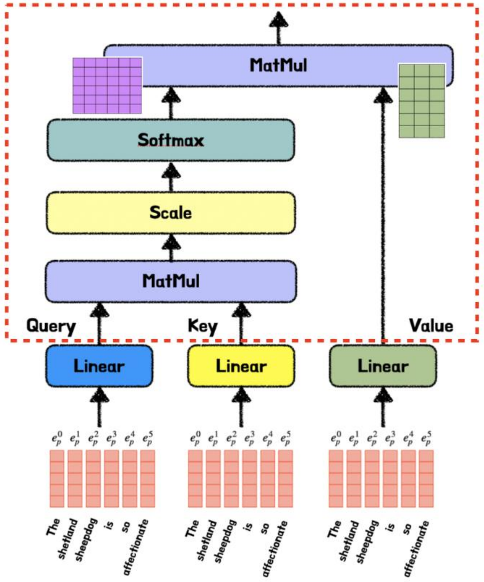


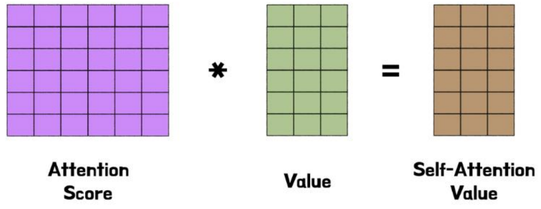


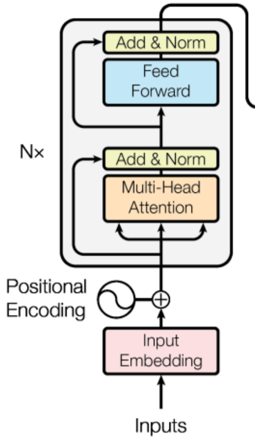
2. Background – partial architecture (Multi-Head Self Attention)



"**Self-attention** obtains attention through the similarity between tokens in the same sentence, and using **multi-head**, the model can capture various types of dependencies between input tokens and handle more complex relationships, resulting in richer expressions."







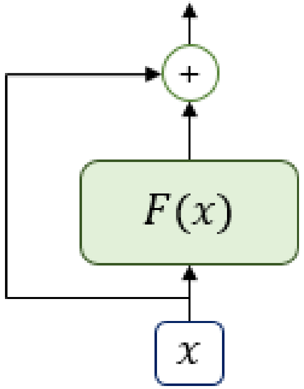
2. Background – partial architecture (Add & Norm)

5. 한눈에 보는 과정

1. **Residual Connection Output**에서 각 벡터 x_i를 뽑음.
2. 각 벡터 x_i에 대해 평균(μ_i)과 분산(σ_i^2) 계산.
3. 평균과 분산으로 정규화하여 $\hat{x}_i$ 계산.
4. γ와 β를 적용하여 최종 출력 ln_i 생성.

Residual Connection

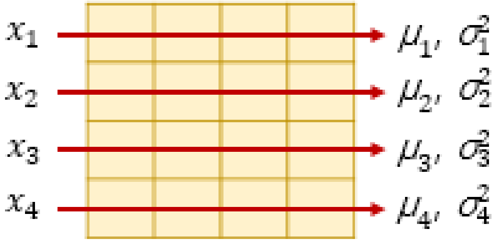
$$H(x) = x + F(x)$$



Multi-head attention output

Multi-head attention input

Layer Normalization

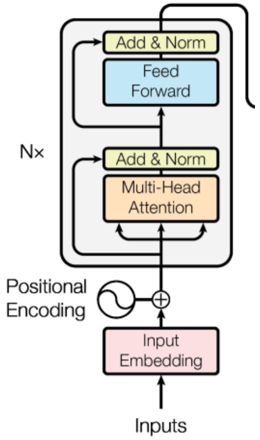


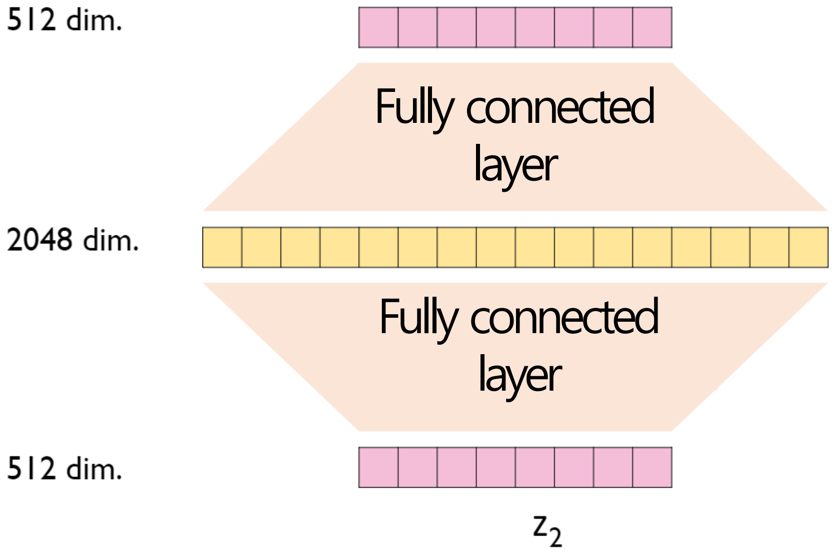
Residual Connection output

$$\hat{x}_{i,k} = \frac{x_{i,k} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}$$

$$ln_i = \gamma \hat{x}_i + \beta = LayerNorm(x_i)$$

2. Background – partial architecture (Feed Forward)





$$F_3 = F_2 W_2 + b_2$$

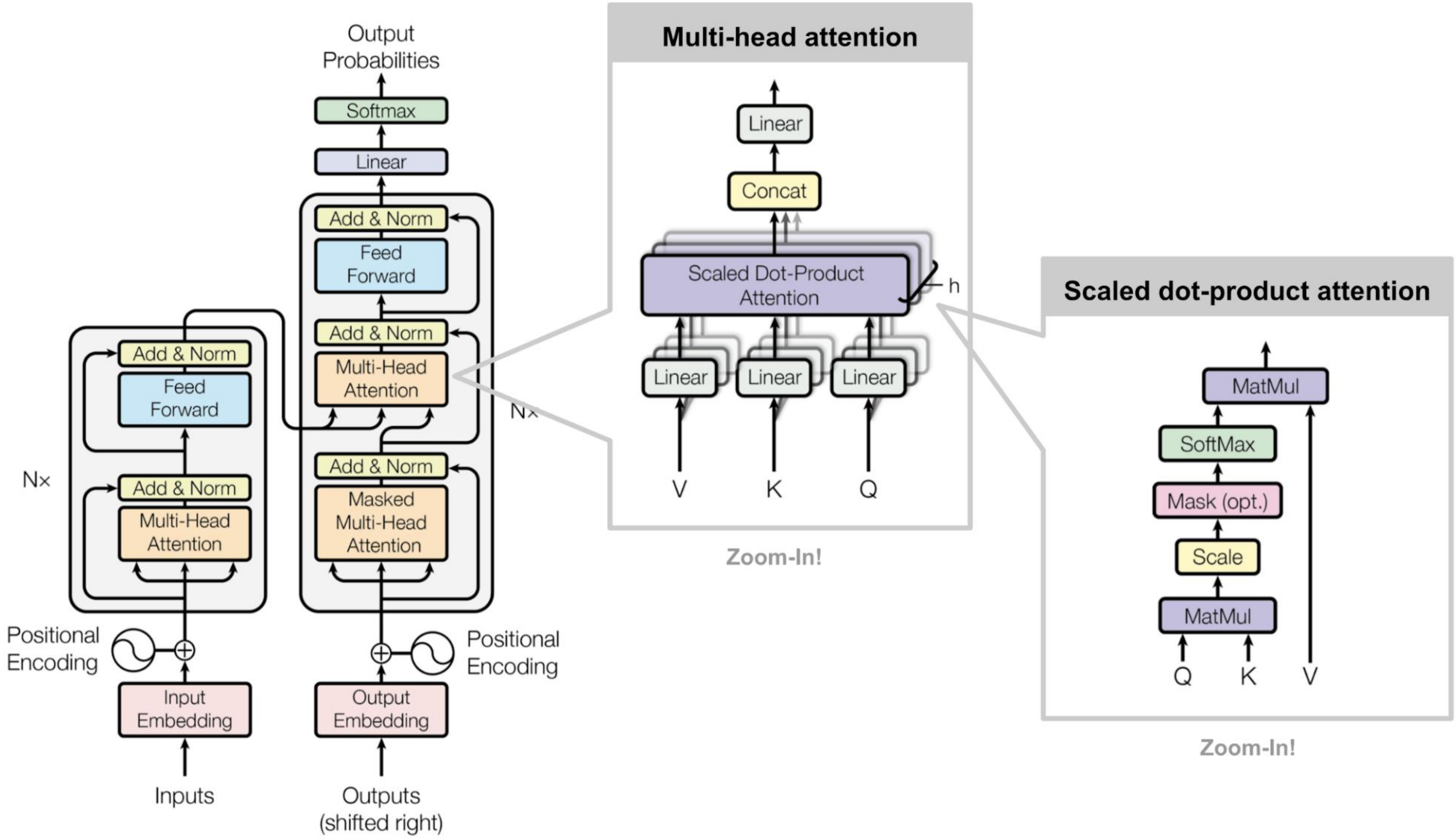
$$F_2 = \mathrm{ReLU}(F_1)$$

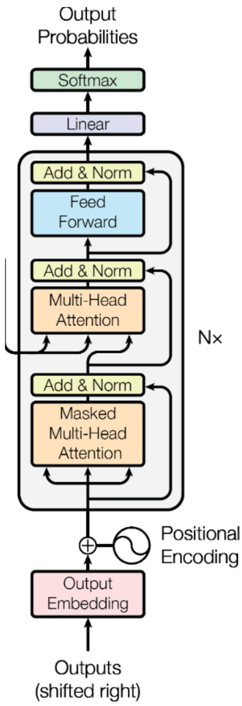
$$F_1 = x\, W_1 + b_1$$

$$< d_{model} \times d_{ff} >$$

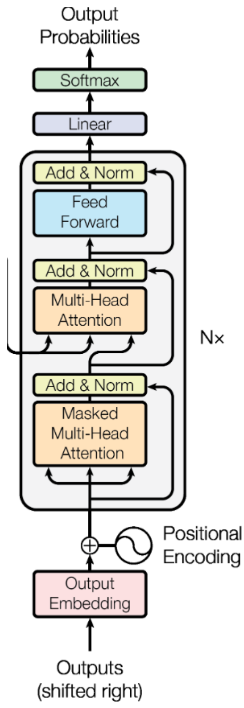
"In the **Feed-Forward** stage, dimensional expansion allows the model to not only learn relationships between words but also enhance the features of individual words. This process improves the model's representational power while maintaining the same input dimensions, aiding in effective learning."

2. Background – partial architecture (Masked Multi-Head Attention)





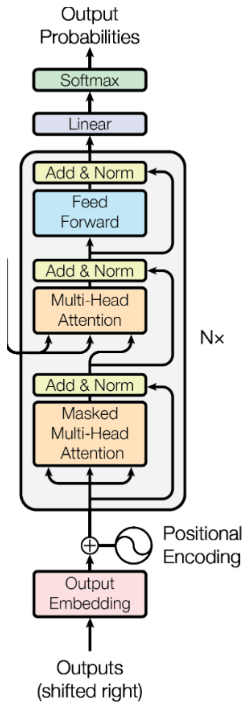
2. Background – partial architecture (Masked Multi-Head Attention)



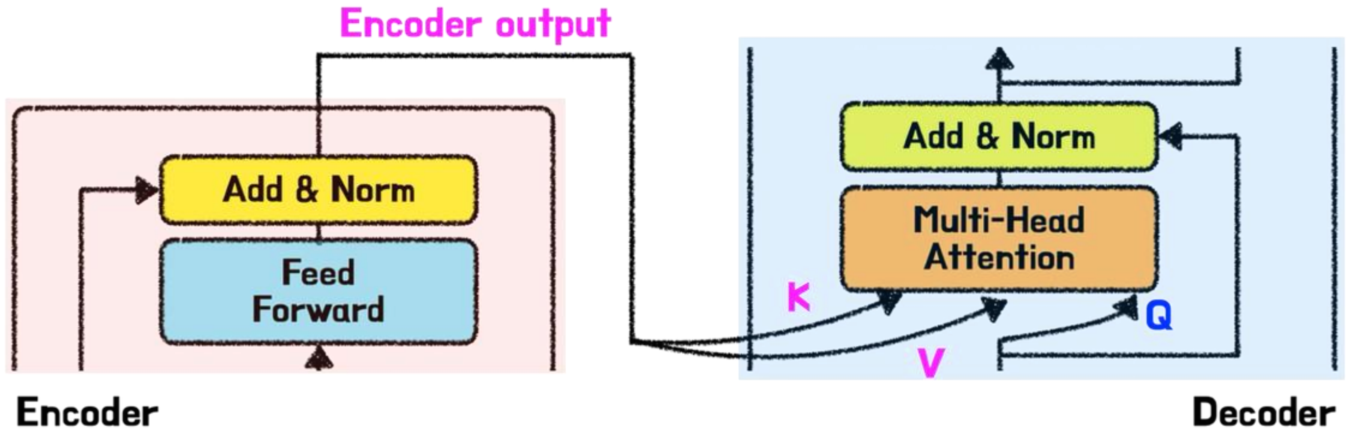
	<sos>	I	am	a	student	<eos>
<sos>		**-inf**	**-inf**	**-inf**	**-inf**	**-inf**
I			**-inf**	**-inf**	**-inf**	**-inf**
am				**-inf**	**-inf**	**-inf**
a					**-inf**	**-inf**
student						**-inf**
<eos>						

“When predicting the current word, information about future words can be used, so the **Mask** technique is used.”

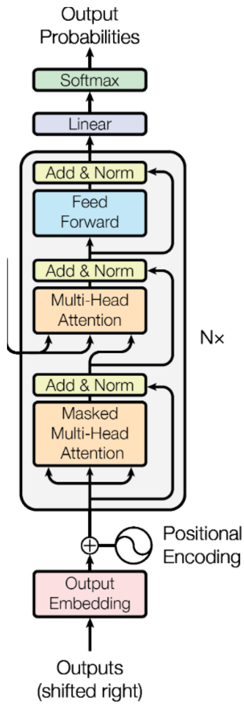
2. Background – partial architecture (Intrance of Encoder-Decoder Attention)



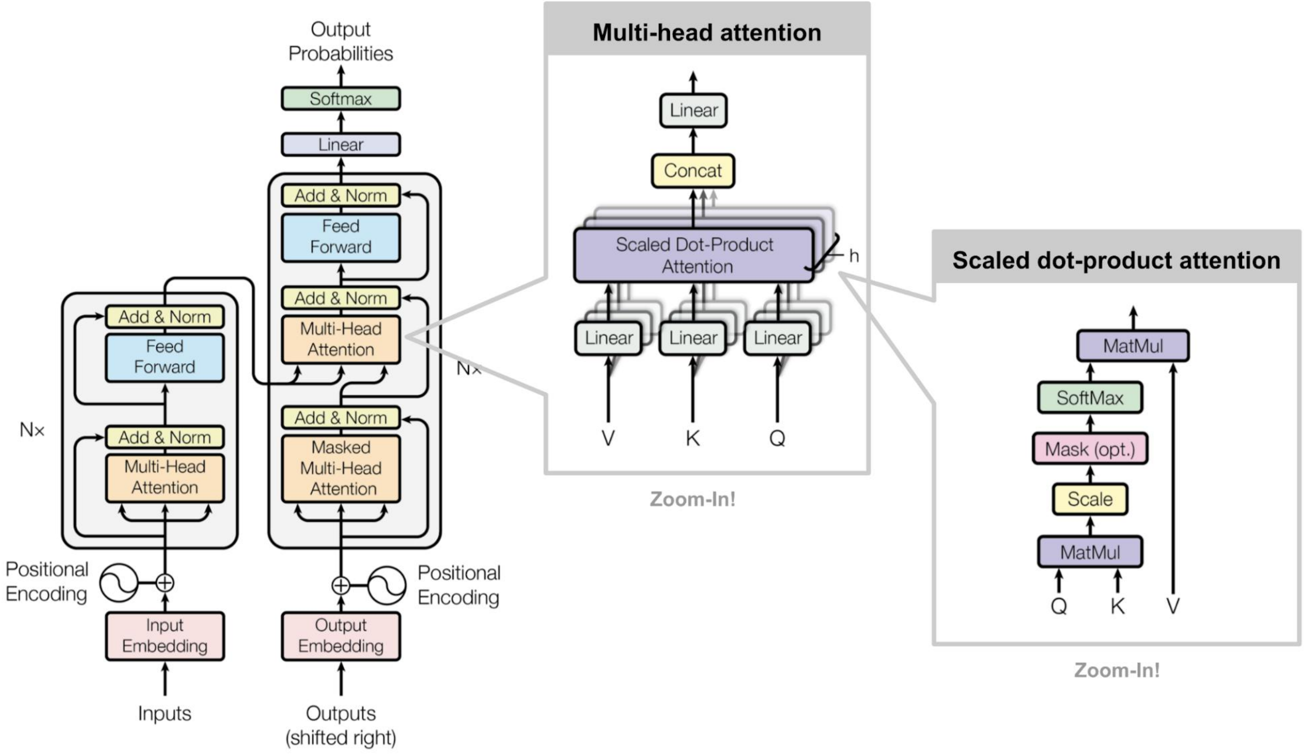
"In Encoder-Decoder Attention, **Q, K, and V are generated from different sources**. Q is generated from the current output of the decoder, and K and V are taken from the output generated from the entire input sequence processed by the encoder."



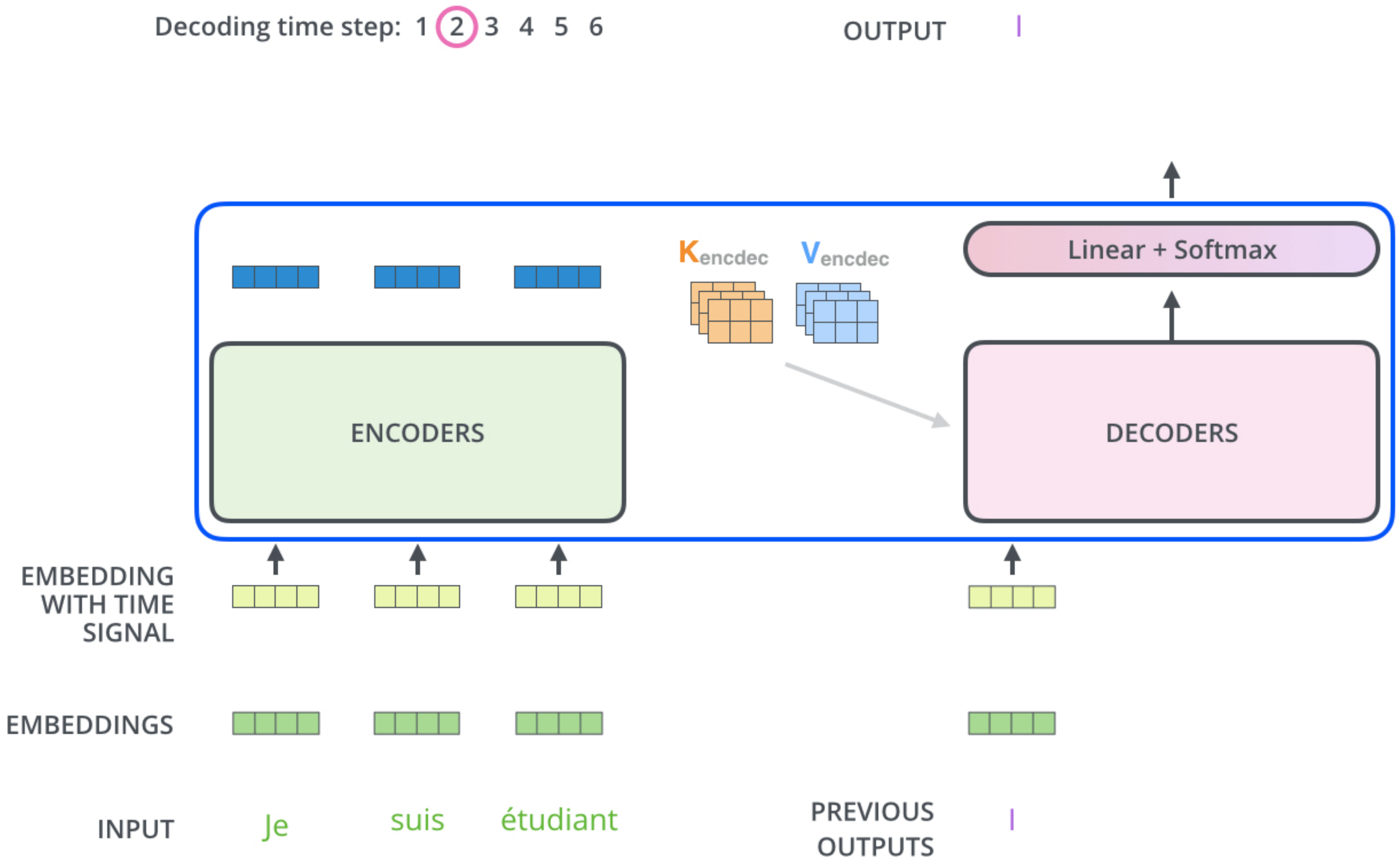
2. Background – partial architecture (Encoder-Decoder Attention)

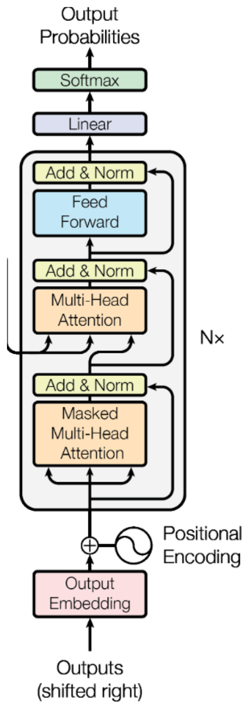


“Thanks to **Encoder-Decoder Attention**, the decoder can calculate which part of the input is most relevant to each output. Based on this, the decoder selectively extracts the necessary information from specific parts of the encoder's output, utilizing the most contextually relevant information at the current step. This process helps the decoder make optimal predictions for the input sentence as a whole.”



2. Background – partial architecture (Visualize model operation process)





2. Results

6 Results

6.1 Machine Translation

On the WMT 2014 English-to-German translation task, the big transformer model (Transformer (big) in Table 2) outperforms the best previously reported models (including ensembles) by more than 2.0 BLEU, establishing a new state-of-the-art BLEU score of 28.4. The configuration of this model is listed in the bottom line of Table 3. Training took 3.5 days on 8 P100 GPUs. Even our base model surpasses all previously published models and ensembles, at a fraction of the training cost of any of the competitive models.

On the WMT 2014 English-to-French translation task, our big model achieves a BLEU score of 41.0, outperforming all of the previously published single models, at less than $1/4$ the training cost of the previous state-of-the-art model. The Transformer (big) model trained for English-to-French used dropout rate $P_{drop} = 0.1$, instead of 0.3.

For the base models, we used a single model obtained by averaging the last 5 checkpoints, which were written at 10-minute intervals. For the big models, we averaged the last 20 checkpoints. We used beam search with a beam size of 4 and length penalty $\alpha = 0.6$ [38]. These hyperparameters were chosen after experimentation on the development set. We set the maximum output length during inference to input length + 50, but terminate early when possible [38].

Table 2 summarizes our results and compares our translation quality and training costs to other model architectures from the literature. We estimate the number of floating point operations used to train a model by multiplying the training time, the number of GPUs used, and an estimate of the sustained single-precision floating-point capacity of each GPU [5].

	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

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

3. Conclusion

7 Conclusion

In this work, we presented the Transformer, the first sequence transduction model based entirely on attention, replacing the recurrent layers most commonly used in encoder-decoder architectures with multi-headed self-attention.

For translation tasks, the Transformer can be trained significantly faster than architectures based on recurrent or convolutional layers. On both WMT 2014 English-to-German and WMT 2014 English-to-French translation tasks, we achieve a new state of the art. In the former task our best model outperforms even all previously reported ensembles.

We are excited about the future of attention-based models and plan to apply them to other tasks. We plan to extend the Transformer to problems involving input and output modalities other than text and to investigate local, restricted attention mechanisms to efficiently handle large inputs and outputs such as images, audio and video. Making generation less sequential is another research goals of ours.

The code we used to train and evaluate our models is available at `https://github.com/tensorflow/tensor2tensor`.

Acknowledgements We are grateful to Nal Kalchbrenner and Stephan Gouws for their fruitful comments, corrections and inspiration.

"The Transformer is the first model to replace recurrent layers with multi-head self-attention, achieving superior performance and faster training speed in tasks like WMT 2014 English-to-German and English-to-French translation. This model has the potential to be extended to handle various inputs and outputs, including not only text but also images and audio."