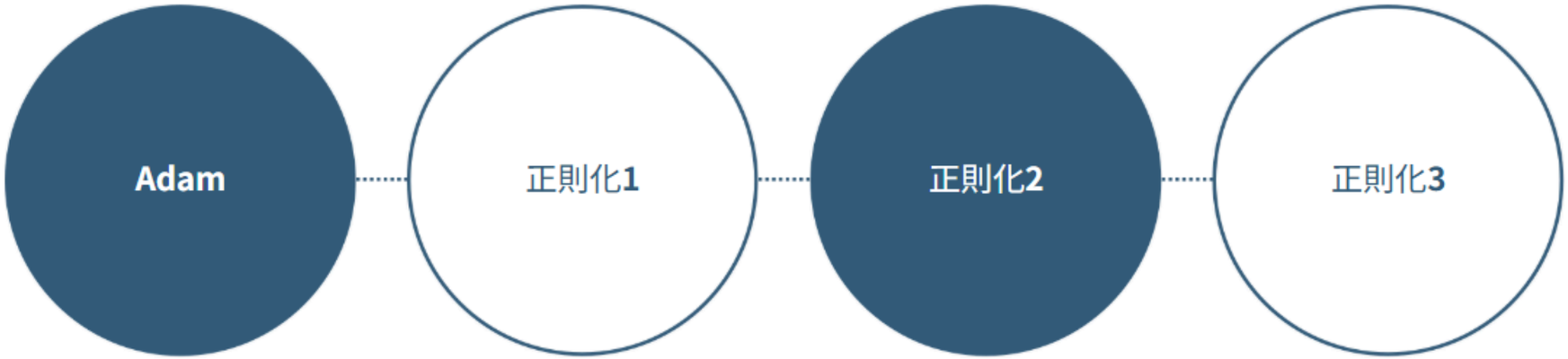
Training



1. モーメントの更新

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$$
$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$$

2. バイアス補正

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

3. パラメータ更新

$$\theta_t = \theta_{t-1} - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

- サブレイヤー出力
- ドロップアウト適用
- 過学習防止

- 埋め込み
位置エンコーディング
合計にドロップアウト

- ラベル平滑化
- 分類精度,BLEUスコアが向上
- perplexityの悪化
-この実験ではϵ= 0.1

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

1. バイトペアエンコーディング（BPE）
言語コーパス中で最も頻出する連続文字列（サブワード）を逐次的にマージし、「頻度が高いサブワード」を語彙に追加していくアルゴリズム。

2. ワードピース（WordPiece）
Google が開発したサブワード分割手法。「マージ後のモデルの尤度向上」を基準にトークンを選びます。

Training

data and batching

ドイツ語ー英語
450万文ペアからなるWMT2014の英語ドイツ語のセット
バイトペアエンコーディングを用いて符号化されており、ソースとターゲットで共有される語彙数は約37,000トークン。

フランス語ー英語
3600万文とトークンを32,000語のワードピース語彙

batch
各訓練バッチには約25,000のソーストークンと約25,000のターゲットトークン

Training

Hardware and Schedule

モデル学習：8 NVIDIA P100 GPUs

Base モデル
ハイパーパラメータは論文内指定のまま
1ステップ ≒ 0.4 秒
合計 100,000 ステップ → 約 0.4 s × 100,000 ≒ 40,000 s ≒ 11.1 h → 論文では約12時間
Big モデル
層数や隠れ層次元などパラメータ数を増やした大型版（Table 3 最下行参照）
1ステップ ≒ 1.0 秒（モデルが大きいため処理が重い）
合計 300,000 ステップ

Result

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**	$2.3 \cdot 10^{19}$	

ドイツ語ー英語

big model→BLUE 28.4

最良のモデル（アンサンブルを含む）を BLEU スコアで 2.0 ポイント以上上回り

合計で約 3.5 日

base model→従来の単体モデルおよびアンサンブルをすべて上回った

フランス語ー英語

big model→BLUE41.0

すべての単体モデルを上回った

学習コストは前例の最先端モデルの約 1/4 未満

ビームサーチの使用

base model→10 分間隔で保存された最後の 5 チェックポイントを平均化した単一モデルを使用

Result

Transformer の各構成要素が性能に与える影響を調べるため，base モデルをいくつかの観点で改変

	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

Result

1. 行 (A): Attention ヘッド数とキー／バリュー次元の変更
2. 計算量を一定に保ちながらヘッド数と d調整。
   - ヘッド数を1つにすると，ベスト設定に比べて約0.9 BLEU 下がる。
   - 一方，多すぎるヘッドでも性能は低下。
3. 行 (B): Attention キー次元 d の縮小
   - d を小さくするとモデル品質が悪化。
   - compatibility判定は容易ではなく，ドット積以上に洗練された関数が有効かもしれないことを示唆。
4. 行 (C),(D): モデルサイズとドロップアウト
   - 大きいモデルほど良い
   - ドロップアウトによる正則化が 過学習防止に極めて有効
5. 行 (E): 位置エンコーディングの違い
   - 正弦・余弦ではなく，「学習可能な位置埋め込み」に置き換えても，
   - base モデルと ほぼ同等の結果。

Result

generalization ablity(見たことないタスクに対応できるか)
英語の構文解析結果

既存モデルのうち Recurrent Neural Network Grammarを除き，すべてを上回る性能を達成
RNN ベースのシーケンス・ツー・シーケンスモデルとは対照的に，WSJ（4万文）のみの訓練でも
BerkeleyParserを超えた

Result

generalization ablity(見たことないタスクに対応できるか)
英語の構文解析結果

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

Result

設定

ハイパーパラメータ

- ドロップアウト（Attention と残差結合両方），学習率，ビーム幅は Section 22(?) の開発セットで微調整
- その他のパラメータは英→独 base 翻訳モデルと同一
- 推論時：最大出力長を「入力長＋300」に設定し，ビーム幅21，長さペナルティ α=0.3 を使用

モデル・データ

- 4 層の Transformer（dmodel=1024）を採用
- 訓練データ：Penn TreebankのWSJ部分（約4万文）
- semi-supervised：高信頼コーパス＋BerkeleyParser コーパスを合わせ約1,700万文を追加
- 語彙サイズ：WSJ のみは16K、半教師付き設定は32K

Conclusion

設定

ハイパーパラメータ

- ドロップアウト（Attention と残差結合両方），学習率，ビーム幅は Section 22(?) の開発セットで微調整
- その他のパラメータは英→独 base 翻訳モデルと同一
- 推論時：最大出力長を「入力長＋300」に設定し，ビーム幅21，長さペナルティ α=0.3 を使用

モデル・データ

- 4 層の Transformer（dmodel=1024）を採用
- 訓練データ：Penn TreebankのWSJ部分（約4万文）
- semi-supervised：高信頼コーパス＋BerkeleyParser コーパスを合わせ約1,700万文を追加
- 語彙サイズ：WSJ のみは16K、半教師付き設定は32K

attention is all you needのmodelと
NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATEのモデル比較

NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATEのmodel構造

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

$$h_j = \left[\overrightarrow{h}_j^\top ; \overleftarrow{h}_j^\top \right]^\top.$$

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

$$p(y_i | y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

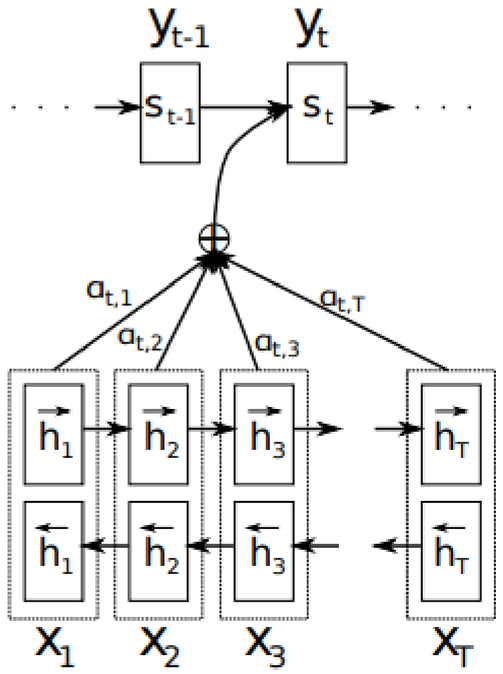


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

attention is all you needのmodelと
NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATEのモデル比較

見解
これら論文を比較すると

状態sにおいて長距離依存の観点から逐次処理しているので
距離が離れるほど情報(理論上はすべての状態を保持しているが)が
遠いほど薄れ、距離に依存する模様。

一方でattentionのみでは各トークンがすべてのトークンを
重みによって参照度合いを変えるので距離に依存しない
優れもの！！