必要なのは注目だけ

アシシュ・ヴァスワニ
Google ブレイン
avaswani@google.com

ノアム・シャジール
Google Brain
noam@google.com

ニキ・パーマー
Google リサーチ
nikip@google.com

ヤコブ・ウスコライト
Google リサーチ
usz@google.com

リオン・ジョーンズ
Google リサーチ
llion@google.com

エイダン・N・ゴメス †トロント大学
aidan@cs.toronto.edu

ukasz Kaiser Google Brain
lukaszkaiser@google.com

イリア・ポロスキン
‡ illia.polosukhin@gmail.com

抽象的な

主要なシーケンス変換モデルは、エンコーダーとデコーダーを含む複雑な再帰型または畳み込み型ニューラルネットワークに基づいています。最もパフォーマンスの高いモデルは、エンコーダーとデコーダーをアテンション メカニズムで接続します。私たちは、再帰と畳み込みを完全に排除し、アテンション メカニズムのみに基づいた新しいシンプルなネットワーク アーキテクチャである Transformer を提案します。2 つの機械翻訳タスクでの実験では、これらのモデルは品質が優れているだけでなく、並列化が可能で、トレーニングにかかる時間が大幅に短縮されることが示されています。私たちのモデルは、WMT 2014 英語からドイツ語への翻訳タスクで 28.4 BLEU を達成し、アンサンブルを含む既存の最高の結果を 2 BLEU 以上上回りました。WMT 2014 英語からフランス語への翻訳タスクでは、私たちのモデルは 8 つの GPU で 3.5 日間トレーニングした後、新しい単一モデルの最先端の BLEU スコア 41.8 を達成しました。これは、文献の最高のモデルのトレーニング コストのほんの一部です。大規模なトレーニング データと限られたトレーニング データの両方を使用して英語の構成構文解析にTransformer をうまく適用することで、Transformer が他のタスクにもうまく一般化できることを示します。

貢献度は同等。リストの順序はランダムです。Jakob は、RNN を自己注意に置き換えることを提案し、このアイデアを評価する取り組みを開始しました。Ashish は、Illia とともに最初の Transformer モデルを設計および実装し、この作業のあらゆる側面に決定的な役割を果たしてきました。Noam は、スケールされたドット積注意、マルチヘッド注意、およびパラメータフリーの位置表現を提案し、ほぼすべての詳細に関わるもう 1 人の人物になりました。Niki は、元のコードベースと tensor2tensor で無数のモデル バリアントを設計、実装、調整、評価しました。Llion も新しいモデル バリアントを試し、初期のコードベースと効率的な推論および視覚化を担当しました。Lukasz と Aidan は、tensor2tensor のさまざまな部分の設計と実装に数え切れないほどの長い時間を費やし、以前のコードベースを置き換えて結果を大幅に改善し、研究を大幅に加速しました。

† Google Brain 在籍中に遂行した業務。
‡ Google Research 在籍中に遂行した業務。

第31回神経情報処理システム会議（NIPS 2017）、米国カリフォルニア州ロングビーチ。

1 はじめに

リカレントニューラルネットワーク、特に長期短期記憶[13]とゲートリカレント[7]ニューラルネットワークは、言語モデルや機械翻訳[35、2、5]などのシーケンスモデリングと変換問題における最先端のアプローチとして確固たる地位を築いてきました。それ以来、リカレント言語モデルとエンコーダーデコーダーアーキテクチャの限界を押し広げるための数多くの取り組みが続けられてきました[38、24、15]。

リカレントモデルは、通常、入力シーケンスと出力シーケンスのシンボル位置に沿って計算を因数分解します。位置を計算時間のステップに合わせると、前の隠れ状態 ht－1 と位置 t の入力の関数として、隠れ状態 ht のシーケンスが生成されます。この本質的に順次的な性質により、トレーニング例内での並列化が不可能になります。これは、メモリ制約により例間のバッチ処理が制限されるため、シーケンスの長さが長くなると重要になります。最近の研究では、因数分解トリック[21]と条件付き計算[32]によって計算効率が大幅に向上し、後者の場合のモデルのパフォーマンスも向上しています。ただし、順次計算の基本的な制約は残っています。

注意メカニズムは、さまざまなタスクにおける強力なシーケンスモデリングと変換モデルの不可欠な部分となり、入力シーケンスまたは出力シーケンス内の距離に関係なく依存関係をモデリングすることを可能にします[2、19]。ただし、いくつかの例外を除いて[27]、このような注意メカニズムは再帰ネットワークと組み合わせて使用されます。

この研究では、再帰を避け、代わりに入力と出力の間のグローバルな依存関係を描画するために注意メカニズムに完全に依存するモデル アーキテクチャである Transformer を提案します。
Transformer は、8 つの P100 GPU でわずか 12 時間のトレーニングで、大幅な並列化を可能にし、翻訳品質の新たな最高水準に到達できます。

2 背景

順次計算を減らすという目標は、Extended Neural GPU [16]、 ByteNet [18] 、ConvS2S [9] の基礎にもなっています。これらはすべて畳み込みニューラルネットワークを基本的な構成要素として使用し、すべての入力位置と出力位置に対して隠れた表現を並列に計算します。これらのモデルでは、任意の 2 つの入力位置または出力位置からの信号を関連付けるために必要な操作の数は、位置間の距離に応じて増加します。ConvS2S の場合は線形、ByteNet の場合は対数的です。これにより、離れた位置間の依存関係を学習することがより困難になります[12]。Transformerでは、これは定数の操作数に削減されますが、アテンション重み付け位置を平均化するため有効解像度が低下するという代償があり、この効果は、セクション 3.2 で説明するように、Multi-Head Attention で打ち消します。

自己注意は、イントラ注意とも呼ばれ、単一のシーケンスの異なる位置を関連付けてシーケンスの表現を計算する注意メカニズムです。自己注意は、読解、抽象要約、テキスト含意、タスクに依存しない文表現の学習など、さまざまなタスクで効果的に使用されています[4、27、28、22]。

エンドツーエンドの記憶ネットワークは、シーケンスに沿った再帰ではなく再帰的な注意メカニズムに基づいており、単純な言語の質問応答や言語モデリングタスクで優れたパフォーマンスを発揮することが示されている[34]。

しかし、私たちの知る限りでは、Transformerは、シーケンス整列RNNや畳み込みを使用せずに、入力と出力の表現を計算するために自己注意に完全に依存している最初の変換モデルです。次のセクションでは、Transformerについて説明し、自己注意の理由を説明し、[17、18]や[9]などのモデルに対する利点について説明します。

3 モデルアーキテクチャ

競合的神経配列伝達モデルのほとんどはエンコーダー-デコーダー構造を持っています[5、2、35] 。
ここで、エンコーダは入力シンボル表現シーケンス（x1、 ...、 xn)を連続表現シーケンスz = （z1、 ...、 zn)にマッピングします。zが与えられると、デコーダは一度に1つの要素のシンボルの出力シーケンス（y1、 ...、 ym)を生成します。各ステップでモデルは自己回帰[10]であり、次のシンボルを生成するときに、以前に生成されたシンボルを追加入力として消費します。

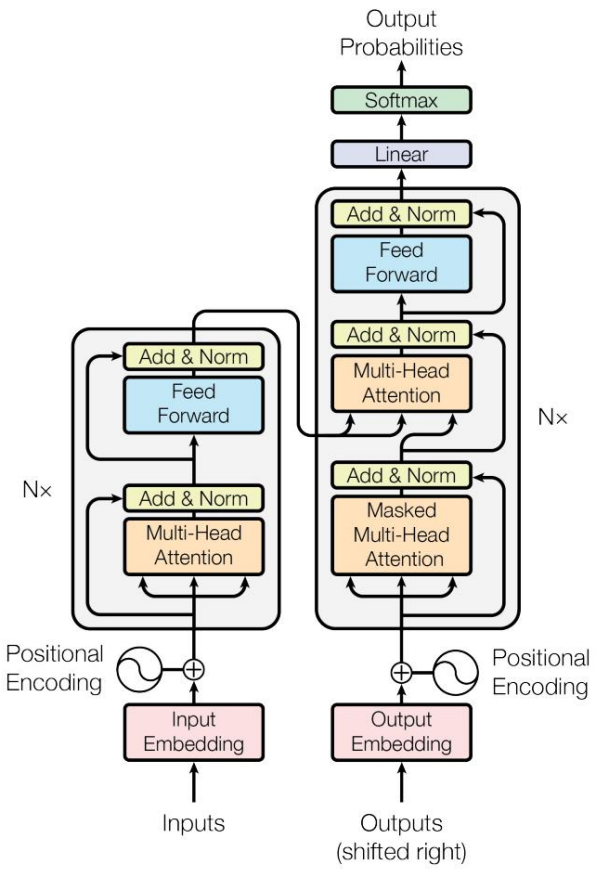


図 1: Transformer - モデル アーキテクチャ。

Transformer は、図 1 の左半分と右半分にそれぞれ示されているように、エンコーダーとデコーダーの両方に対して、積み重ねられた自己注意とポイントごとの完全接続レイヤーを使用して、この全体的なアーキテクチャに従います。

3.1 エンコーダーとデコーダーのスタック

エンコーダー:エンコーダーは、 N = 6の同一レイヤーのスタックで構成されています。各レイヤーには 2 つのサブレイヤーがあります。1 つ目はマルチヘッド セルフアテンション メカニズムで、2 つ目は位置ごとに完全に接続された単純なフィードフォワード ネットワークです。2 つのサブレイヤーのそれぞれに残差接続[11]を使用し、その後レイヤー正規化[1]を実行します。つまり、各サブレイヤーの出力はLayerNorm(x + Sublayer(x)) です。ここで、 Sublayer(x)はサブレイヤー自体によって実装される関数です。これらの残差接続を容易にするために、モデル内のすべてのサブレイヤーと埋め込みレイヤーは、次元dmodel = 512 の出力を生成します。

デコーダー:デコーダーもN = 6 個の同一レイヤーのスタックで構成されています。各エンコーダー レイヤーの 2 つのサブレイヤーに加えて、デコーダーは 3 番目のサブレイヤーを挿入し、エンコーダー スタックの出力に対してマルチヘッド アテンションを実行します。エンコーダーと同様に、各サブレイヤーの周囲に残差接続を使用し、その後レイヤーの正規化を行います。また、デコーダー スタックの自己アテンション サブレイヤーを変更して、位置が後続の位置に注意を向けないようにします。このマスキングと、出力埋め込みが 1 つの位置だけオフセットされるという事実を組み合わせることで、位置 i の予測は、i より小さい位置の既知の出力のみに依存するようになります。

3.2 注意

アテンション関数は、クエリとキーと値のペアのセットを出力にマッピングするものと説明できます。クエリ、キー、値、出力はすべてベクトルです。出力は加重合計として計算されます。

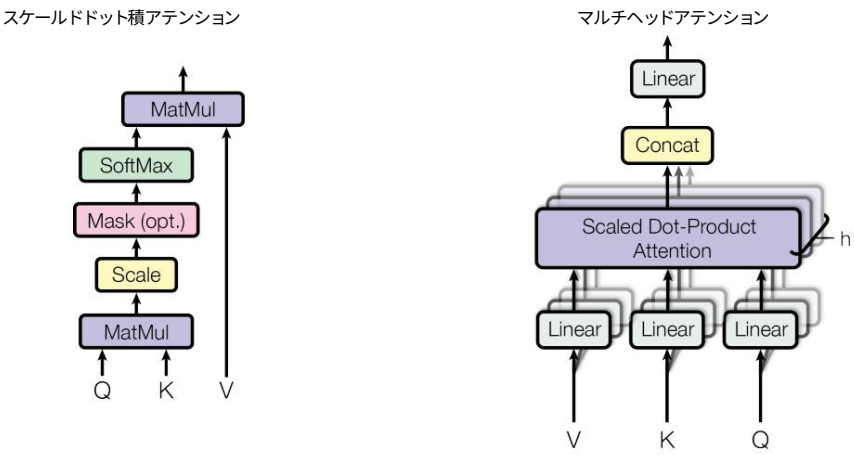


図 2: (左) スケールされたドット積アテンション。 (右) マルチヘッド アテンションは、並行して実行される複数のアテンション レイヤーで構成されます。

各値に割り当てられる重みは、対応するキーを持つクエリの互換性関数によって計算されます。

3.2.1 スケールドドット積アテンション

私たちは、この特別な注意を「スケールド ドット積注意」と呼んでいます (図 2)。入力は、次元dkのクエリとキー、および次元dv の値で構成されます。クエリとすべてのキーのドット積を計算し、それぞれを√dkで割り、ソフトマックス関数を適用して値の重みを取得します。

実際には、クエリのセットに対して同時にアテンション関数を計算し、行列Q にまとめます。キーと値も行列KとVにまとめられます。出力の行列は次のように計算します。

$$\text{注意}(Q, K, V) = \text{ソフトマックス}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (1)$$

最も一般的に使用される2つのアテンション関数は、加法アテンション[2]とドット積（乗法)アテンションです。ドット積アテンションは、スケーリング係数を除いて、私たちのアルゴリズムと同じです。加法アテンションは、単一の隠れ層を持つフィードフォワードネットワークを使用して互換性関数を計算します。この2つは理論的な複雑さは似ていますが、ドット積アテンションは、高度に最適化された行列乗算コードを使用して実装できるため、実際にははるかに高速でスペース効率に優れています。

dkの値が小さい場合、 2つのメカニズムは同様に機能しますが、 dkの値が大きい場合、加法的な注意はスケーリングなしでドット積の注意よりも優れています[3]。dkの値が大きい場合、ドット積の値が大きくなり、ソフトマックス関数が極端に小さな勾配を持つ領域に押し込まれると考えられます。

[4]。この効果を打ち消すために、ドット積を√dkでスケールする。

3.2.2 マルチヘッドアテンション

dmodel次元のキー、値、クエリで単一の注意機能を実行する代わりに、クエリ、キー、値をそれぞれdk、 dk 、 dv次元に異なる学習済み線形投影でh回線形投影すると効果的であることがわかりました。クエリ、キー、値の投影された各バージョンで、注意機能を並列に実行し、 dv次元を生成します。

4ドット積が大きくなる理由を説明するために、 qとkの成分が独立したランダムqikiであり、平均が0で分散がdkであると仮定し平均0 、分散1の変数。そのドット積q・k =

出力値。これらは連結され、再度投影され、図 2 に示すように最終値になります。

マルチヘッド アテンションにより、モデルは異なる位置にある異なる表現サブスペースからの情報に共同で注意を向けることができます。単一のアテンション ヘッドでは、平均化によってこれが妨げられます。

MultiHead(Q, K, V) = Concat(head1, ..., headh)WO、ここでheadi =

Attention(QWQ は、こんにちけ私、 Vウェストバージニア州私

ここで、投影はパラメータ行列WとWO∈Rhdv ×dmodelである。 質問私 ∈ R dモデル×dk 、で け私 ∈ R dモデル×dk 、WV私∈ R dmodel×dv

。

この研究では、 h = 8 個の並列アテンション レイヤー、つまりヘッドを使用します。これらのそれぞれに対して、 dk = dv = dmodel/h = 64を使用します。各ヘッドの次元が削減されるため、計算コストの合計は、フル次元の単一ヘッド アテンションの計算コストと同程度になります。

3.2.3 モデルにおける注意の応用

Transformer は、マルチヘッド アテンションを 3 つの異なる方法で使用します。

- 「エンコーダ-デコーダアテンション」層では、クエリは前のデコーダ層から取得され、メモリキーと値はエンコーダの出力から取得されます。これにより、デコーダのすべての位置が入力シーケンスのすべての位置を監視できます。これは、 [38、2、9]などのシーケンスツーシーケンスモデルの一般的なエンコーダ-デコーダアテンションメカニズムを模倣しています。

- エンコーダーには自己注意層が含まれています。自己注意層では、すべてのキー、値、クエリが同じ場所、この場合はエンコーダーの前の層の出力から取得されます。エンコーダー内の各位置は、エンコーダーの前の層のすべての位置に注意を向けることができます。

- 同様に、デコーダーの自己注意層により、デコーダーの各位置が、その位置までを含むデコーダー内のすべての位置に注意を向けることができます。自己回帰特性を維持するには、デコーダー内で左向きの情報の流れを防ぐ必要があります。これをスケールされたドット積注意の内部で実装するには、ソフトマックスの入力で不正な接続に対応するすべての値をマスクします (-∞ に設定) 。図 2 を参照してください。

3.3 位置ごとのフィードフォワードネットワーク

注意サブレイヤーに加えて、エンコーダーとデコーダーの各レイヤーには、各位置に個別に同一に適用された完全に接続されたフィードフォワード ネットワークが含まれています。これは、間に ReLU アクティベーションを挟んだ 2 つの線形変換で構成されます。

$$FFN(x) = \text{最大}(0, xW1 + b1)W2 + b2 \quad (2)$$

線形変換は位置が異なっても同じですが、レイヤーごとに異なるパラメータを使用します。これを別の方法で説明すると、カーネル サイズが 1 の 2 つの畳み込みとなります。
入力と出力の次元はdmodel = 512 で、内部層の次元はdf f = 2048 です。

3.4 埋め込みとソフトマックス

他のシーケンス変換モデルと同様に、学習した埋め込みを使用して、入力トークンと出力トークンを次元dmodelのベクトルに変換します。また、通常の学習した線形変換とソフトマックス関数を使用して、デコーダー出力を予測された次のトークンの確率に変換します。私たちのモデルでは、[30]と同様に、2つの埋め込み層とプレソフトマックス線形変換の間で同じ重み行列を共有します。埋め込み層では、それらの重みに√dmodelを掛けます。

表1: 最大パス長、レイヤーごとの複雑さ、および最小連続操作数
異なるレイヤータイプの場合。nはシーケンスの長さ、 dは表現次元、 kはカーネル
畳み込みのサイズと r は制限付き自己注意における近傍のサイズです。

レイヤータイプ	レイヤーごとの複雑さ	連続最大パス長 オペレーション	
自己注意	の上 ^{2} ・	オー(1)	オー(1)
再発	O(n・d ^{2} d))	の上)	の上)
畳み込み	O(k・n・d ^{2})	オー(1)	O(log(n))
自己注意（制限あり）	または(r・n・d)	オー(1)	O(n/r)

3.5 位置エンコーディング

このモデルには再帰や畳み込みが含まれていないため、モデルが
シーケンスの順序を変更するには、シーケンスの相対的または絶対的な位置に関する情報を挿入する必要があります。
シーケンス内のトークン。この目的のために、入力埋め込みに「位置エンコーディング」を追加します。
エンコーダとデコーダスタックの底部。位置エンコーディングは同じ次元dmodelを持つ。
埋め込みとして、2つを足し合わせることができる。位置エンコーディングには多くの選択肢があり、
学習され、修正された[9]。

この研究では、異なる周波数の正弦関数と余弦関数を使用します。

P E(pos,2i) = sin(pos/100002i/dmodel)

P E(pos,2i+1) = cos(pos/100002i/dmodel)

ここでposは位置、 iは次元である。つまり、位置エンコーディングの各次元は
正弦波に対応します。波長は2πから10000•2πまでの等比数列を形成します。
この関数を選んだのは、モデルが簡単に出席を学習できると仮定したためである。
相対位置は、任意の固定オフセットkに対して、 P Epos+kは線形関数として表すことができるため、
Pエポス。

また、代わりに学習した位置埋め込み[9]を使用する実験も行い、2つの
どちらのバージョンもほぼ同じ結果となった（表3の行（E)を参照）。我々は正弦波バージョンを選択した。
モデルが遭遇したシーケンスの長さよりも長いシーケンスの長さに外挿できるようにする可能性があるため
トレーニング中。

4 なぜ自己注意が必要なのか

このセクションでは、自己注意層のさまざまな側面を、 1つの可変長シンボル表現シーケンスをマッピングするために一般的に使用される再帰層と畳み込み層と比較します。
(x1, ..., xn)を、同じ長さの別のシーケンス(z1, ..., zn)に変換します。ここで、 xiは 、zi∈R ^{d} 、隠された
典型的なシーケンス変換エンコーダまたはデコーダの層。自己注意の使用の動機は、
3つの願いを考えてみましょう。

1つはレイヤーあたりの計算量の合計です。もう1つは、
必要な連続操作の最小数によって測定されるように並列化されます。

3つ目は、ネットワーク内の長距離依存関係間のパスの長さです。長距離学習
依存関係は、多くの配列伝達タスクにおける重要な課題です。
このような依存関係を学習する能力は、前方および後方の信号が到達しなければならない経路の長さである。
ネットワークを横断する。入力内の任意の位置の組み合わせ間のこれらのパスが短いほど、
出力シーケンスが長いほど、長距離依存性を学習しやすくなります[12]。したがって、
ネットワーク内の任意の2つの入力位置と出力位置間の最大経路長は、
さまざまなレイヤー タイプ。

表1に示すように、自己注意層は、すべての位置を一定数の連続した
実行される操作はO(n)回であるのに対し、再帰層ではO(n)回の連続した操作が必要となる。
計算の複雑さを考えると、自己注意層は再帰層よりも高速である。

長さnは表現の次元dよりも小さく、これは単語ピース[38]やバイトペア[31]表現など、機械翻訳の最先端のモデルで使用される文表現で最も頻繁に当てはまります。非常に長いシーケンスを含むタスクの計算パフォーマンスを向上させるために、自己注意は、それぞれの出力位置を中心とした入力シーケンス内のサイズrの近傍のみを考慮するように制限できます。これにより、最大パス長がO(n/r)に増加します。今後の研究では、このアプローチをさらに調査する予定です。

カーネル幅k < nの単一の畳み込み層では、入力位置と出力位置のすべてのペアが接続されるわけではありません。これを行うには、連続カーネルの場合はO(n/k)畳み込み層のスタックが必要になり、拡張畳み込み[18]の場合はO(logk(n))となり、ネットワーク内の任意の 2 つの位置間の最長パスの長さが増加します。畳み込み層は一般に、再帰層よりも k 倍高価です。ただし、分離可能畳み込み [6] では、複雑性が大幅に削減され、 O(k・n・d + n・d畳み込みは、自己注意層とポイントごとのフィードフォワード層の組み合わせに等しく、このモデルで採用しているアプローチです。
^{2}）。しかし、 k = nであっても、分離可能な

副次的な利点として、自己注意によって、より解釈しやすいモデルが生み出される可能性があります。私たちは、モデルからの注意分布を調べ、付録で例を示して議論します。個々の注意ヘッドは明らかに異なるタスクを実行することを学習するだけでなく、多くは文の統語的および意味的構造に関連する動作を示すようです。

5 トレーニング

このセクションでは、モデルのトレーニング方法を説明します。

5.1 トレーニングデータとバッチ処理

約 450 万の文ペアからなる標準 WMT 2014 英語 - ドイツ語データセットでトレーニングしました。文はバイトペアエンコーディング[3] を使用してエンコードされ、約 37,000 トークンの共有ソース -ターゲット語彙を持っています。英語 - フランス語については、3600 万の文で構成されるかなり大規模な WMT 2014 英語 - フランス語データセットを使用し、トークンを 32,000 語の語彙に分割しました[38]。文ペアは、おおよそのシーケンス長ごとにバッチ処理されました。各トレーニングバッチには、約 25,000 のソーストークンと 25,000 のターゲットトークンを含む文ペアのセットが含まれていました。

5.2 ハードウェアとスケジュール

私たちは、8 つの NVIDIA P100 GPU を搭載した 1 台のマシンでモデルをトレーニングしました。論文全体で説明されているハイパーパラメータを使用したベース モデルの場合、各トレーニング ステップに約 0.4 秒かかりました。ベース モデルは合計 100,000 ステップ、つまり 12 時間トレーニングしました。大きなモデル (表 3 の一番下の行に記載) の場合、ステップ時間は 1.0 秒でした。大きなモデルは 300,000 ステップ(3.5 日)トレーニングされました。

5.3 オプティマイザ

Adamオプティマイザー[20]をβ1 = 0.9、 β2 = 0.98 、 = 10－9で使用しました。学習率は次の式に従ってトレーニング中に変化させました。

$$\text{lrate} = d_{\text{モデル}}^{-0.5} \cdot \min(\text{ステップ数}^{-0.5}, \text{ステップ数} \cdot \text{ウォームアップステップ数}^{-1.5}) \quad (3)$$

これは、最初のwarmup_stepsトレーニング ステップでは学習率を直線的に増加し、その後はステップ数の逆平方根に比例して学習率を減少させることに相当します。warmup_steps = 4000 を使用しました。

5.4 正規化

トレーニング中は、次の 3 種類の正規化を採用します。

表2: Transformerは、これまでの最先端モデルよりも優れたBLEUスコアを達成しています。
英語からドイツ語、英語からフランス語への newstest2014 テストを、わずかなトレーニング コストで実施できます。

モデル	青		トレーニングコスト (FLOP)	
	EN-DE	EN-FR 23.75	EN-DE	EN-FR
バイトネット [18]				
ディープアタック + ポスアンク [39]		39.2		1.0 · 1020
GNMT + RL [38]	24.6	39.92	2.3 · 1019	1.4 · 1020
コンバージドS2S [9]	25.16	40.46	9.6 · 1018	1.5 · 1020
文部科学省 [32]	26.03	40.56	2.0 · 1019	1.2 · 1020
Deep-Att + PosUnk アンサンブル [39]		40.4		8.0 · 1020
GNMT + RLアンサンブル[38]	26.30	41.16	1.8 · 1020	1.1 · 1021
ConvS2Sアンサンブル[9]	26.36	41.29	7.7 · 1019	1.2 · 1021
トランスフォーマー（ベースモデル）	27.3	38.1	3.3 · 1018	
トランスフォーマー（大）	28.4	41.8	2.3 · 1019	

残差ドロップアウト各サブレイヤーの出力にドロップアウト[33]を適用し、それを
サブレイヤー入力を正規化する。さらに、埋め込みと
エンコーダーとデコーダースタックの両方で位置エンコーディングを使用します。基本モデルでは、
Pドロップ=0.1。

ラベルスムージングトレーニング中、 ls = 0.1 [36]の値のラベルスムージングを採用した。
モデルが不確実性を高めることを学習するため、困惑度は低下しますが、精度と BLEU スコアは向上します。

6 件の結果

6.1 機械翻訳

WMT 2014の英語からドイツ語への翻訳タスクでは、大きなトランスフォーマーモデル（Transformer (big)
表2のモデルは、これまでに報告された最高のモデル（アンサンブルを含む）よりも2.0以上優れている。
BLEUは、28.4という最新のBLEUスコアを確立しました。このモデルの構成は
表3の一番下の行に記載されています。トレーニングには8つのP100 GPUで3.5日かかりました。ベースモデルでも
これまでに公開されたすべてのモデルとアンサンブルを凌駕し、トレーニングコストは
競合モデル。

WMT 2014の英語からフランス語への翻訳タスクでは、当社の大規模モデルはBLEUスコア41.0を達成しました。
これまでに発表されたすべての単一モデルよりも優れた性能を、トレーニングコストの1/4以下で実現しました。
以前の最先端のモデル。英語からフランス語への翻訳用に訓練されたTransformer（big)モデルは、
ドロップアウト率Pdrop = 0.3 ではなく 0.1 です。

ベースモデルには、最後の5つのチェックポイントの平均から得られた単一のモデルを使用しました。
10分間隔で書き込まれました。大きなモデルでは、最後の20のチェックポイントを平均しました。
ビームサイズ4 、長さペナルティα = 0.6のビームサーチを使用した[38]。これらのハイパーパラメータ
開発セットでの実験後に選択された。
入力長+50までの推論を行うが、可能な場合は早期に終了する[38]。

表2は、私たちの結果をまとめ、翻訳品質とトレーニングコストを他のモデルと比較したものです。
文献からアーキテクチャを推定します。
トレーニング時間、使用したGPUの数、持続時間の推定値を掛け合わせてモデルを作成します。
各GPUの単精度浮動小数点容量[5]。

6.2 モデルのバリエーション

Transformerのさまざまなコンポーネントの重要性を評価するために、ベースモデルを変更しました。
さまざまな方法で、英語からドイツ語への翻訳のパフォーマンスの変化を測定し、

5K80、K40、M40、P100 にはそれぞれ 2.8、3.7、6.0、9.5 TFLOPS の値を使用しました。

表3: Transformerアーキテクチャのバリエーション。記載されていない値は基本値と同一です。
モデル。すべての指標は英語からドイツ語への翻訳開発セットnewstest2013に掲載されています。
困惑は、バイトペアエンコーディングに従って単語ごとに発生し、
単語ごとの困惑。

	N dモデル	dff h dk dv Pdrop	ls		PPL BLEUパラメータをトレーニングする ステップ (dev) (dev) 25.8 24.9		×106
6進数	512 2048 8 64 64 0.1			0.1 100K 4.92 5.29 5.00			65
（あ）		1 512 512 4 128			4.91		
		128 16 32 32 32			5.01	25.5	
		16 16				25.8	
						25.4	
（バ）		16			5.16	25.1	58
		32			5.01	25.4	60
（ハ）	2				6.11	23.7	36
	4				5.19	25.3	50
	8				4.88	25.5	80
	256	32 32 128			5.75	24.5	28
	1024	128			4.66	26.0	168
	1024				5.12	25.4	53
	4096				4.75	26.2	90
（デ）			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) 正弦波の代わりに位置埋め込み 大きい 6 1024 4096 16					4.92	25.7	
			0.3	300K 4.33		26.4	213

開発セット、newstest2013。前のセクションで説明したようにビームサーチを使用しましたが、
チェックポイントの平均化。これらの結果を表3に示す。

表3の行（A)では、注目ヘッドの数と注目キーと値の次元を変えています。
3.2.2節で説明したように、計算量を一定に保つ。シングルヘッドの場合、
注意は最良設定よりも 0.9 BLEU 悪く、ヘッドが多すぎると品質も低下します。

表3の行（B)では、注目キーサイズdkを小さくするとモデルの品質が低下することがわかります。
互換性を判断するのは簡単ではなく、より洗練された互換性が求められることを示唆している。
関数よりもドット積の方が有益である可能性がある。さらに、行（C)と（D)では、予想通り、
モデルが大きいほど良く、ドロップアウトは過剰適合を避けるのに非常に役立ちます。行（E)では、
学習された位置埋め込みを用いた正弦波位置エンコーディング[9]では、ほぼ同一の
結果をベースモデルに反映します。

6.3 英語構成構文解析

Transformerが他のタスクにも応用できるかどうかを評価するため、英語で実験を行った。
選挙区解析。このタスクには特定の課題があります。出力は強い構造的影響を受けます。
制約があり、入力よりも大幅に長くなります。さらに、RNNシーケンスツーシーケンス
モデルは小規模データ領域では最先端の結果を達成することができなかった[37]。

我々はdmodel = 1024の4層トランスフォーマーをウォールストリートジャーナル（WSJ)の部分で訓練した。
ペンツリーバンク[25]、約4万の訓練文。また、半教師あり設定で訓練した。
約1700万文の大規模で信頼性の高いBerkleyParserコーパスを使用して

[37] WSJのみの設定では16Kトークンの語彙を使用し、WSJのみの設定では32Kトークンの語彙を使用した。
半教師あり設定の場合。

我々は、注目度と残差の両方のドロップアウトを選択するために、少数の実験を行った。
（セクション5.4）、セクション22開発セットの学習率とビームサイズ、その他すべてのパラメータ
英語からドイツ語への基本翻訳モデルから変更はありませんでした。推論中に、

表4: Transformerは英語の構成構文解析にうまく一般化します（結果はセクション23にあります）

（ウォールストリートジャーナル）

パーサー	トレーニング	WSJ 23 F1
Vinyals & Kaiser el al. (2014) [37] WSJのみ、識別WSJのみ、識別WSJのみ、識別WSJのみ、識別		88.3
ペトロフら（2006年)[29]	半教師あり半教師あり半教師あり半教師あり半教師あり半教師ありマルチタスク生成	90.4
Zhu et al. （2013年)[40]		90.4
ダイアーら（2016)[8]		91.7
トランスフォーマー（4層）		91.3
Zhu et al. （2013年)[40]		91.3
黄•ハーパー（2009)[14]		91.3
マクロスキーら（2006)[26]		92.1
Vinyals & Kaiser の。 （2014年)[37]		92.1
トランスフォーマー（4層）		92.7
Luongら（2015年)[23]		93.0
ダイアーら（2016)[8]		93.3

最大出力長を入力長+300に増加しました。ビームサイズは21 、 α = 0.3を使用しました。
WSJのみと半教師あり設定の両方について。

表4の結果は、タスク固有のチューニングが不足しているにもかかわらず、私たちのモデルが驚くほどうまく機能し、

リカレントニューラルネットワーク文法[8]

RNNシーケンスツーシーケンスモデル[37]とは対照的に、 Transformerは40K文のWSJトレーニングセットのみでトレーニングした場合でも、 Berkeley-Parser[29]よりも優れたパフォーマンスを発揮します。

7 結論

この研究では、完全に
エンコーダ•デコーダアーキテクチャで最も一般的に使用される再帰層を
多頭自己注意。

翻訳タスクの場合、Transformerはアーキテクチャベースのものよりも大幅に速くトレーニングできます。
再帰層または畳み込み層。WMT 2014英語からドイツ語とWMT 2014の両方で
英語からフランス語への翻訳タスクでは、私たちは新たな最先端技術を達成しました。前者のタスクでは、私たちの最高の
このモデルは、これまでに報告されたすべてのアンサンブルよりも優れています。

私たちは注目度ベースのモデルの将来に興奮しており、それを他のタスクに適用する予定です。
Transformerをテキスト以外の入出力モダリティを含む問題に拡張する計画と
大規模な入力と出力を効率的に処理するための局所的かつ制限された注意メカニズムを調査する
画像、音声、動画などです。生成の連続性を減らすことも私たちの研究目標の 1 つです。

モデルのトレーニングと評価に使用したコードはhttps://github.com/で入手できます。
tensorflow/tensor2tensor。

謝辞有益な情報を提供してくれたNal Kalchbrenner氏とStephan Gouws氏に感謝します。
コメント、訂正、インスピレーション。

参考文献

[1]ジミー•レイ•バ、ジェイミー•ライアン•キロス、ジェフリー•E•ヒントン。レイヤー正規化。arXivプレプリント
arXiv:1607.06450、 2016年。

[2] Dzmitry Bahdanau、Kyunghyun Cho、Yoshua Bengio。共同研究によるニューラル機械翻訳
整列と翻訳の学習。CoRR、abs/1409.0473、2014。

[3]デニー•ブリッツ、アンナ•ゴールディ、ミン•タン•ルオン、クオック•V•レ。神経細胞の大規模探索
機械翻訳アーキテクチャ。CoRR、abs/1703.03906、2017。

[4]江鵬鵬、李東、ミレラ•ラパタ「機械のための長期短期記憶ネットワーク」
読書。arXivプレプリントarXiv:1601.06733、 2016年。

[5] Kyunghyun Cho、Bart van Merrienboer、Caglar Gulcehre、Fethi Bougares、Holger Schwenk、 Yoshua Bengio。統計的機械翻訳のためのRNNエンコーダー•デコーダーを使用したフレーズ表現の学習。CoRR、abs/1406.1078、2014年。

[6]フランソワ•ショレ。Xception: 深さ方向に分離可能な畳み込みによる深層学習。arXiv プレプリントarXiv:1610.02357、 2016年。

[7] Junyoung Chung、Çaglar Gülçehre、Kyunghyun Cho、Yoshua Bengio。シーケンスモデリングにおけるゲート付きリカレントニューラルネットワークの実証的評価。CoRR、abs/1412.3555、2014。

[8] Chris Dyer、Adhiguna Kuncoro、Miguel Ballesteros、Noah A. Smith。リカレントニューラルネットワーク文法。NAACL紀要、2016年。

[9] Jonas Gehring、Michael Auli、David Grangier、Denis Yarats、Yann N. Dauphin。畳み込みシーケンスからシーケンスへの学習。arXivプレプリントarXiv :1705.03122v2、 2017年。

[10]アレックス•グレイブス。リカレントニューラルネットワークによるシーケンス生成。arXivプレプリント arXiv:1308.0850、 2013年。

[11] Kaiming He、Xiangyu Zhang、Shaoqing Ren、Jian Sun。画像認識のための深層残差学習。IEEE Computer Vision and Pattern Recognitionカンファレンス論文集、770〜778ページ、2016年。

[12]ゼップ•ホッホライター、ヨシュア•ベンジオ、パオロ•フラスコーニ、ユルゲン•シュミットフーバー。勾配の流れ リカレントネット :長期依存関係を学習することの難しさ、2001 年。

[13]ゼップ•ホッホライターとユルゲン•シュミットフーバー。長期短期記憶。ニューラルコンピューティング、9(8):1735–1780, 1997年。

[14]黄中強、メアリー•ハーパー。言語間での潜在的注釈によるPCFG文法の自己学習。2009年自然言語処理における経験的手法に関する会議の議事録、832〜841ページ。ACL、2009年8月。

[15] Rafal Jozefowicz、Oriol Vinyals、Mike Schuster、Noam Shazeer、Yonghui Wu。言語モデルの限界を探る。arXivプレプリントarXiv:1602.02410、 2016年。

[16] ukasz KaiserとSamy Bengio。能動記憶は注意力に取って代わることができるか ?Advances in Neural 情報処理システム（NIPS）、2016年。

[17] ukasz KaiserとIlya Sutskever。ニューラルGPUはアルゴリズムを学習する。国際学習表現会議（ICLR）、2016年。

[18] Nal Kalchbrenner、Lasse Espeholt、Karen Simonyan、Aaron van den Oord、Alex Graves、Ko-ray Kavukcuoglu。線形時間でのニューラル機械翻訳。arXivプレプリントarXiv :1610.10099v2、 2017年。

[19] Yoon Kim、Carl Denton、Luong Hoang、Alexander M. Rush。構造化注意ネットワーク。 2017年国際学習表現会議にて。

[20] Diederik KingmaとJimmy Ba. Adam :確率的最適化のための手法。ICLR、2015年。

[21] Oleksii KuchaievとBoris Ginsburg。LSTMネットワークの因数分解トリック。arXivプレプリントarXiv:1703.10722、 2017年。

[22] Zhouhan Lin、Minwei Feng、Cicero Nogueira dos Santos、Mo Yu、Bing Xiang、Bowen Zhou、Yoshua Bengio。構造化された自己注意文の埋め込み。arXivプレプリントarXiv :1703.03130、 2017年。

[23] Minh-Thang Luong、Quoc V. Le、Ilya Sutskever、Oriol Vinyals、およびLukasz Kaiser。マルチタスクシーケンスツーシーケンス学習。arXivプレプリントarXiv :1511.06114、 2015年。

[24] Minh-Thang Luong、Hieu Pham、Christopher D Manning。注意に基づくニューラル機械翻訳への効果的なアプローチ。 arXivプレプリントarXiv :1508.04025、 2015年。

[25]ミッチェル•P•マーカス、メアリー•アン•マルチンキエヴィッチ、ベアトリス•サントリーニ。英語の大規模注釈付きコーパスの構築：ペン•ツリーバンク。計算言語学、19(2):313–330、1993年。

[26]デビッド•マクロスキー、ユージン•チャーニアック、マーク•ジョンソン「構文解析のための効果的な自己トレーニング」 NAACL人間言語技術会議メインカンファレンスの議事録、 152〜159ページ。ACL、2006年6月。

[27] Ankur Parikh、Oscar Täckström、Dipanjan Das、Jakob Uszkoreit。分解可能な注意モデル。自然言語処理における経験的手法、2016年。

[28]ロマン•パウルス、カイミン•シオン、リチャード•ソッチャー。抽象的思考のための深層強化モデル要約。arXivプレプリントarXiv:1705.04304、 2017年。

[29] Slav Petrov、Leon Barrett、Romain Thibaux、Dan Klein。正確でコンパクト、かつ解釈可能なツリー注釈の学習。第21回国際計算言語学会議および第44回ACL年次会議の議事録、433〜440ページ。ACL、 2006年7月。

[30] Ofir PressとLior Wolf。出力埋め込みを用いた言語モデルの改善。arXivプレプリントarXiv:1608.05859、 2016年。

[31]リコ•ゼンリッチ、バリー•ハドウ、アレクサンドラ•バーチ。サブワード単位による希少単語のニューラル機械翻訳。arXivプレプリントarXiv:1508.07909、 2015年。

[32] Noam Shazeer、Azalia Mirhoseini、Krzysztof Maziarz、Andy Davis、Quoc Le、Geoffrey Hinton、 Jeff Dean。とてつもなく大きなニューラルネットワーク：スパースゲートされた専門家混合層。arXivプレプリントarXiv：1701.06538、 2017年。

[33] Nitish Srivastava、Geoffrey E Hinton、Alex Krizhevsky、Ilya Sutskever、Ruslan Salakhutdinov 。ドロップアウト：ニューラルネットワークの過剰適合を防ぐ簡単な方法。機械学習研究ジャーナル、15(1):1929–1958、2014年。

[34]サインバヤル•スフバータル、アーサー•シュラム、ジェイソン•ウェストン、ロブ•ファーガス。エンドツーエンドのメモリネットワーク。C.コルテス、NDローレンス、DDリー、M.杉山、R.ガーネット編、神経情報処理システムの進歩28、2440-2448ページ。カランアソシエイツ社、2015年。

[35] Ilya Sutskever、Oriol Vinyals、Quoc VV Le。ニューラルネットワークによるシーケンスツーシーケンス学習。Advances in Neural Information Processing Systems、3104-3112ページ、2014年。

[36]クリスチャン•セゲディ、ヴィンセント•ヴァンホーク、セルゲイ•イオフェ、ジョナソン•シュレンス、ズビグネフ•ヴォイナ。コンピュータービジョンのインセプションアーキテクチャの再考。CoRR、abs/1512.00567、2015。

[37] Vinyals & Kaiser, Koo, Petrov, Sutskever、Hinton。外国語としての文法。Advances in Neural Information Processing Systems、2015年。

[38] Yonghui Wu、Mike Schuster、Zhifeng Chen、Quoc V Le、Mohammad Norouzi、Wolfgang Macherey、Maxim Krikun、Yuan Cao、Qin Gao、Klaus Macherey、他「Googleのニューラル機械翻訳システム：人間翻訳と機械翻訳のギャップを埋める」arXivプレプリントarXiv：1609.08144、 2016年。

[39] Jie Zhou、Ying Cao、Xuguang Wang、Peng Li、Wei Xu。ニューラル機械翻訳のための高速フォワード接続を備えたディープリカレントモデル。CoRR、abs / 1606.04199、2016年。

[40] Muhua Zhu、Yue Zhang、Wenliang Chen、Min Zhang、Jingbo Zhu。高速で正確なシフトリデュース構成要素解析。ACL第51回年次会議の議事録（第1巻：長編論文）、434〜443ページ。ACL、2013年8月。

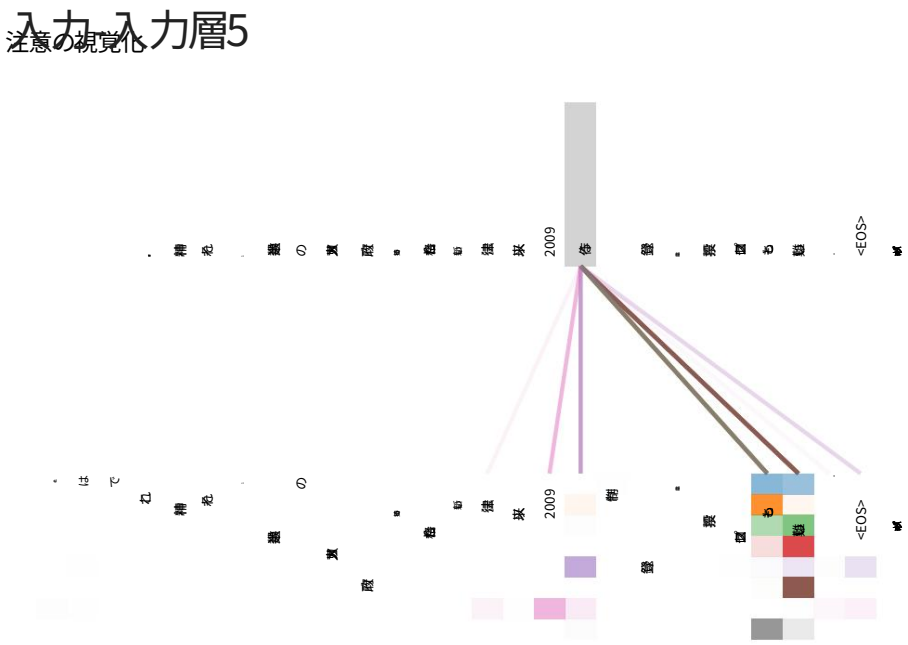


図 3: 6 層のうち 5 層目のエンコーダ自己注意における長距離依存関係に従う注意メカニズムの例。注意ヘッドの多くは、動詞「making」の遠隔依存関係に注意を向け、「making...more difficult」というフレーズを完成させます。ここでは、単語「making」のみに注意が向けられています。異なる色は異なるヘッドを表します。カラーで表示すると最適です。

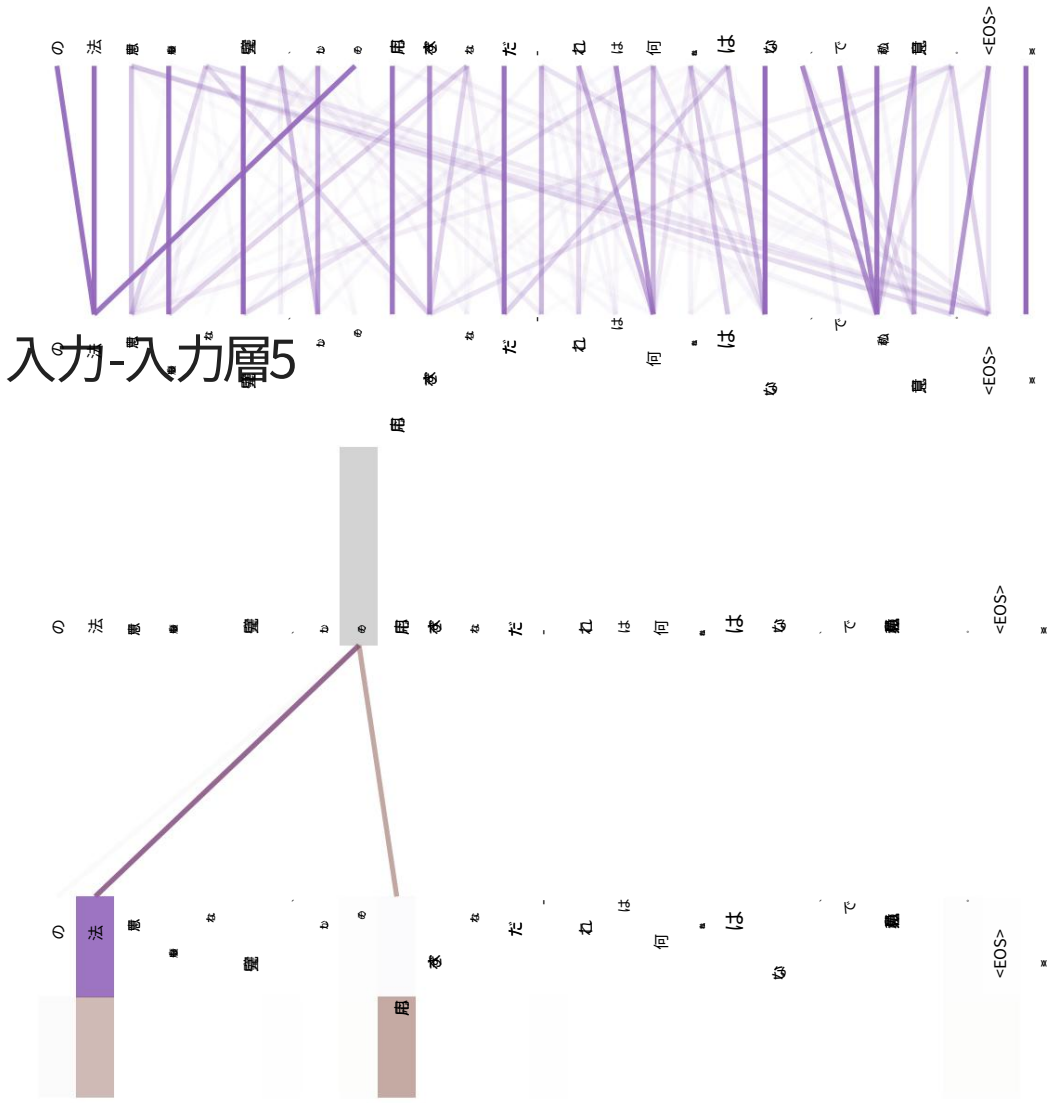
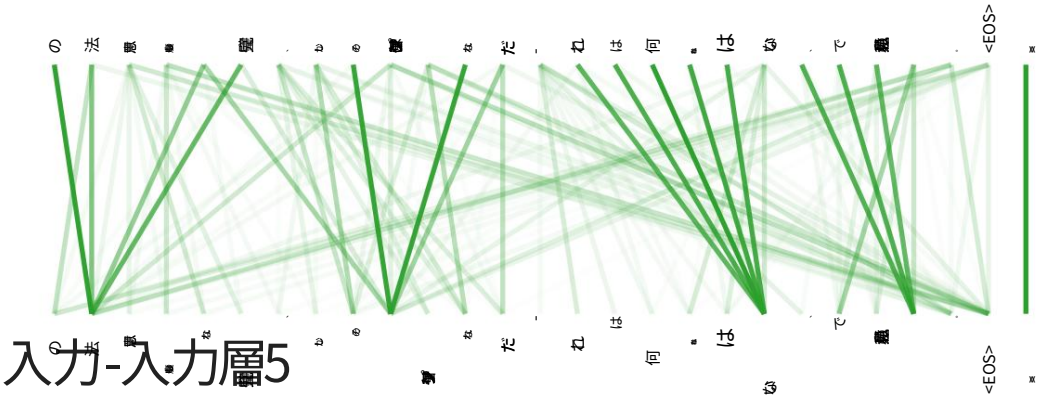


図 4: 2 つの注意ヘッド (レイヤー 5/6 に存在) は、明らかにアナフォラ解決に関与しています。上:ヘッド 5 の完全な注意。下: 注意ヘッド 5 と 6 の単語「its」のみからの分離された注意。この単語に対する注意が非常に鋭いことに注意してください。



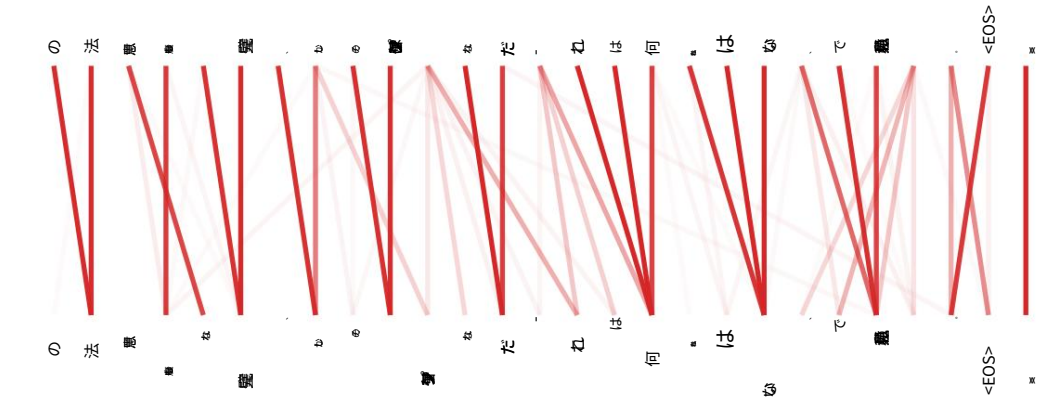


図 5: 多くの注意ヘッドは、文の構造に関連していると思われる動作を示します。上には、6 層のうち 5 層目のエンコーダ自己注意からの 2 つの異なるヘッドからの 2 つの例を示します。ヘッドは明らかに異なるタスクを実行することを学習しました。