言語モデルのパラメータから探る Detokenization メカニズム

鴨田豪 [1] Benjamin Heinzerling[2,1] 稲葉達郎 [3] 工藤慧音 [1,2]
坂口慶祐 [1,2] 乾健太郎 [4,1,2]
[1] 東北大学 [2] 理化学研究所 [3] 京都大学 [4]MBZUAI
go.kamoda@dc.tohoku.ac.jp benjamin.heinzerling@riken.jp
inaba@sap.ist.i.kyoto-u.ac.jp keito.kudo.q4@dc.tohoku.ac.jp
keisuke.sakaguchi@tohoku.ac.jp kentaro.inui@mbzuai.ac.ae

概要

トークナイザは単語を複数のサブワードに分割することがあるが，その分割が言語的に意味のあるものになるとは限らない．推論の段階仮説（Stages of inference hypothesis）では，言語モデルの序盤層はこうしたサブワードトークン列をより意味のある表現に変換 (Detokenize) するとされている．本研究では，従来のプロービングや因果介入などの経験的手法に依存せず，Detokenization をモデルの重みに基づく解析によって観測できることを示す．具体的には，GPT-2 の第 1 層の注意機構を解析的に分解し，トークンタイプに由来する寄与とトークンの位置に由来する項の寄与とを切り分けた分析を行い，近いトークンや頻出 Bigram への注意の偏りを明らかにする[1)]．

1 はじめに

近年の多くの言語モデル (LM) [1–5] はサブワードトークン [6,7] を入出力に用いる．そのため，LM は “Sapiens” のような単語や名前を，“Sap” と “iens” のように部分に分割された形で扱うことがある．このようにトークン分割（Tokenization）は必ずしも言語的に意味のあるものになるとは限らず，言語モデルの初期層は，こうしたサブワードトークン列をより意味のある単語や名前の表現に変換する **Detokenization** の役割を持つとされている [8]．Detokenization に関して，これまではモデルへの入力文の選択やプローブの訓練を伴う経験的な実験がなされ，どの層が Detokenization に関連する振る舞いを示すかが示されてきた [9,10]．本研究では，GPT-2 [2] の第 1 層の注意機構を分解することで，Detokenization の重要な側面のいくつかが経験的な手法を用いずともモデルの重みから理解できることを示す．

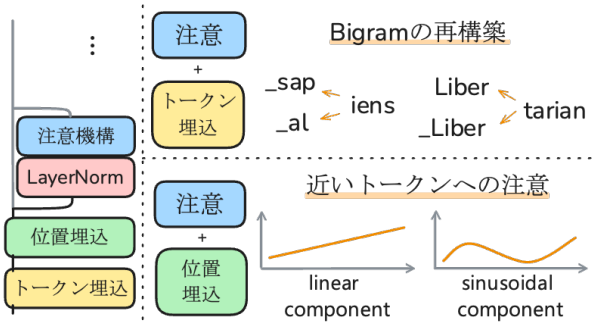

図 1 トークン埋込，位置埋込，LayerNorm と第 1 注意機構に着目し，モデルの重みを分析する．モデル計算を分解することで，Detokenization の 2 つの重要な条件である Bigram の再構築と近いトークンへの注意を分離して解釈する．

Detokenization において重要なのは，単語やフレーズを構成するトークンに対する注意である．これまでの研究では，Detokenization の n-gram に対する注意を分析してきた [9,10]．しかし，これらの分析はトークン埋め込みによる影響と位置埋め込みによる影響を分離していない．本研究では，LayerNorm を考慮しながら，位置情報を除いた表現の注意に対する影響を分析する (§ 3)．

もう一つの重要な側面は，近いトークンへの注意である．本研究では重みに対する分析を行い，GPT-2 の第 1 層では，入力トークンに関係なく，位置的に近いトークンに高い注意が割り当てられることを示す（§ 4）．さらに，学習された絶対位置埋め込みに 2 つの成分から構成されることを示す．第 1 の成分は，ALiBi [11] に類似した線形バイアス成分と見なすことができる．第 2 の成分は，正弦波形状を持ち，正弦波 [1] やロータリー [12] 位置エンコーディングを思い起こさせる．これらの 2 つの成分が

1) github.com/gokamoda/lm-detokenization

重なり合うことで，近いトークンに対する注意の偏りが生じていることを示す．

2 第 1 層注意機構の分解

本研究では，Lad らの Detokenization や注意の傾向を調査する先行研究 [8] に倣い，GPT-2 を分析対象とする．GPT-2 は，モデルの解釈可能性に焦点を当てた他の先行研究でも分析対象となっている [13–15]．

重みの分析を行う前に，モデルのいくつかの重みや関数を再定義して分析を簡単にする．続く節では，GPT-2 の LayerNorm，注意機構に焦点を当て，複数の線形変換を一つの線形変換に折り込み，無視できる項が存在することを示す．

2.1 埋め込み層

GPT-2 の初期の隠れ状態は，トークンの ID と絶対位置に基づいて計算される．ここで，i 番目のトークンの ID を ID_i，語彙を V，埋め込みの次元数を d，トークン埋め込み行列を $\boldsymbol{E} \in \mathbb{R}^{|V|\times d}$ とする．さらに，GPT-2 は絶対位置エンコーディングを採用しており，モデルが受け付ける最大トークン長を L，位置エンコーディング行列を $\boldsymbol{P} \in \mathbb{R}^{L\times d}$ とすると，位置 i の埋め込み層の出力 $\boldsymbol{x}_i$ は次のように表せる：

$$\boldsymbol{x}_i = \boldsymbol{e}_{\mathrm{ID}_i} + \boldsymbol{p}_i \tag{1}$$

2.2 LayerNorm

Transformer アーキテクチャは，さまざまなポイントで層正規化を適用する．GPT-2 で使用されている層正規化 (LayerNorm) [2] は，次のように表される：

$$\underline{\mathrm{LN}}(\boldsymbol{x}) := \frac{\boldsymbol{x}}{\sigma(\boldsymbol{x})}\left(\boldsymbol{I} - \frac{1}{d}\mathbf{1}^\top\mathbf{1}\right)\mathrm{diag}(\boldsymbol{\gamma}) + \boldsymbol{\beta} \tag{2}$$

$$\sigma(\boldsymbol{x}) := \sqrt{\mathrm{Var}(\boldsymbol{x}) + \epsilon} \tag{3}$$

ここで，ϵ はゼロ除算を防ぐために追加される小さな定数であり，$\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$ は学習可能なパラメータである．したがって，LayerNorm は $\sigma(\boldsymbol{x})$ による除算以外の部分は線形なアフィン変換である．

2.3 注意機構

注意機構の役割は，文脈情報を現在のトークンの表現に動的に混ぜ込むことである．現在の位置 i と文脈情報 $\boldsymbol{X}$ が与えられた場合，Decoder モデルの H 個のヘッドを持つ注意層は次の計算を行う：

$$\mathrm{ATTN}(i, \boldsymbol{X}) := \sum_{h=1}^{H}\sum_{j=1}^{i} \alpha_{i,j,h}\boldsymbol{v}_h(\boldsymbol{x}_j)\boldsymbol{W}_h^O + \boldsymbol{b}^O \tag{4}$$

ここで，$\boldsymbol{v}_h(\boldsymbol{x}_j)$ は，$\boldsymbol{x}_j$ を次元 $d' = d/H$ の Value ベクトルに変換するアフィン変換，行列 $\boldsymbol{W}^O$ とベクトル $\boldsymbol{b}^O$ は出力アフィン変換の重みとバイアスであり，トークン位置 i から j へのヘッド h による注意の重み $\alpha_{i,j,h}$ は次のように与えられる：

$$\alpha_{i,j,h} := \underset{j\le i}{\mathrm{softmax}}\left(s_{i,j,h}/\sqrt{d'}\right) \tag{5}$$

$$\underline{s_{i,j,h}} := \boldsymbol{q}_h(\boldsymbol{x}_i)\boldsymbol{k}_h(\boldsymbol{x}_j)^\top \tag{6}$$

ここで，$\sum_j \alpha_{i,j,h} = 1$ が成り立ち，$s_{i,j,h}$ は正規化されていない注意スコアである．また，$\boldsymbol{q}_h$ と $\boldsymbol{k}_h$ は入力 $\boldsymbol{x}$ から Query ベクトルと Key ベクトルへのアフィン変換である：

$$\boldsymbol{q}_h(\boldsymbol{x}) := \boldsymbol{x}\boldsymbol{W}_h^Q + \boldsymbol{b}_h^Q \tag{7}$$

$$\boldsymbol{k}_h(\boldsymbol{x}) := \boldsymbol{x}\boldsymbol{W}_h^K + \boldsymbol{b}_h^K \tag{8}$$

2.4 LayerNorm と注意機構の再定義

式 4, 5 から，ATTN の計算では，入力 $\boldsymbol{x}$ は最初に必ずアフィン変換されることがわかる．ATTN への入力は LN の出力であるため，$\underline{\mathrm{LN}}$の線形部分は，ATTN の $\boldsymbol{q}_h$，$\boldsymbol{k}_h$，$\boldsymbol{v}_h$ 関数のアフィン変換に吸収させることができる．これを踏まえて，式 2 の代わりに，LN を再定義する：

$$\mathrm{LN}(\boldsymbol{x}) := \boldsymbol{x}/\sigma(\boldsymbol{x}) \tag{9}$$

更に，式 7 の $\boldsymbol{q}_h$ で適用されるアフィン変換を再定義し，$\boldsymbol{k}_h$ と $\boldsymbol{v}_h$ に対しても同様の再定義を行う：

$$\boldsymbol{W}_h^Q := \left(\boldsymbol{I} - \frac{1}{d}\mathbf{1}^\top\mathbf{1}\right)\mathrm{diag}(\boldsymbol{\gamma})\underline{\boldsymbol{W}_h^Q} \tag{10}$$

$$\boldsymbol{b}_h^Q := \boldsymbol{\beta}\underline{\boldsymbol{W}_h^Q} + \underline{\boldsymbol{b}_h^Q} \tag{11}$$

次に，式 6 で定義された正規化前の注意スコア $\underline{s_{i,j,h}}$ に着目する：

$$\underline{s_{i,j,h}} = \boldsymbol{q}_h(\boldsymbol{x}_i)\boldsymbol{W}_h^{K\top}\boldsymbol{x}_j^\top + \boldsymbol{q}_h(\boldsymbol{x}_i)\boldsymbol{b}_h^{K\top} \tag{12}$$

式 5 では，softmax がトークン位置 j に適用されているが，式 6 を展開した式 12 の第 2 項は j に依存しない．Softmax は定数の加算に対して不変であるため，この項は無視できる．これらを考慮して更に式 12 を展開すると，$\boldsymbol{W}_h^{QK} := \boldsymbol{W}_h^Q\boldsymbol{W}_h^{K\top}$, $\boldsymbol{b}_h^{QK} := \boldsymbol{b}_h^Q\boldsymbol{W}_h^{K\top}$ を用いて次のように表される：

$$s_{i,j,h} := \boldsymbol{x}_i\boldsymbol{W}_h^{QK}\boldsymbol{x}_j^\top + \boldsymbol{b}_h^{QK}\boldsymbol{x}_j^\top \tag{13}$$

2) 元の定式化からの記号を区別するために，§ 2.4 での再定義された記号を下線で示す

2.5 分解

式 13 の第 1 項は，現在のトークン $\boldsymbol{x}_i$ と過去のトークン $\boldsymbol{x}_j$ の隠れ状態に依存する．2 つの隠れ状態 $\boldsymbol{x}_i$ と $\boldsymbol{x}_j$ が線形射影 $\boldsymbol{W}_h^{QK}$ の下で類似しているときに大きくなるため，この項を「比較項」と呼ぶことにする．第 2 項は，過去のトークン $\boldsymbol{x}_j$ の隠れ状態にのみ依存し，現在のトークン i には依存しない．大きな $\boldsymbol{b}_h^{QK}\boldsymbol{x}_j^\top$ 値は，トークン j が文脈に関係なく重要であることを主張していると捉えられるため，以降この項を「自己主張項」と呼ぶ．

更に式 13 を，Token 埋め込みと位置埋め込みを考慮して展開すれば，モデルの入力から第 1 層の注意機構の出力までの計算は $\sigma_i := \sigma(\boldsymbol{e}_{\mathrm{ID}_i} + \boldsymbol{p}_i)$ を用いて次のように表される:

$$
\begin{aligned}
s_{i,j,h} = {} & \underbrace{\frac{\boldsymbol{e}_{\mathrm{ID}_i}\boldsymbol{W}_h^{QK}\boldsymbol{e}_{\mathrm{ID}_j}^\top}{\sigma_i\sigma_j}}_{T^{\mathrm{ee}}_{i,j,h}} + \underbrace{\frac{\boldsymbol{p}_i\boldsymbol{W}_h^{QK}\boldsymbol{p}_j^\top}{\sigma_i\sigma_j}}_{T^{\mathrm{pp}}_{i,j,h}} + \underbrace{\frac{\boldsymbol{p}_i\boldsymbol{W}_h^{QK}\boldsymbol{e}_{\mathrm{ID}_j}^\top}{\sigma_i\sigma_j}}_{T^{\mathrm{pe}}_{i,j,h}} \\
& + \underbrace{\frac{\boldsymbol{e}_{\mathrm{ID}_i}\boldsymbol{W}_h^{QK}\boldsymbol{p}_j^\top}{\sigma_i\sigma_j}}_{T^{\mathrm{ep}}_{i,j,h}} + \underbrace{\frac{\boldsymbol{b}_h^{QK}\boldsymbol{e}_{\mathrm{ID}_j}^\top}{\sigma_j}}_{T^{\mathrm{e}}_{j,h}} + \underbrace{\frac{\boldsymbol{b}_h^{QK}\boldsymbol{p}_j^\top}{\sigma_j}}_{T^{\mathrm{p}}_{j,h}}
\end{aligned}
\tag{14}
$$

以降，各項は青字で示した記号で参照する．

3 Detokenization とトークン関連度

トークン由来の比較項 T^{ee} は，現在のトークン $\boldsymbol{e}_{\mathrm{ID}_i}$ と過去のトークン $\boldsymbol{e}_{\mathrm{ID}_j}$ の埋め込みを線形変換 $\boldsymbol{W}_h^{QK}$ を介して比較するため，Detokenization に非常に関連していると考えられる．大きな T^{ee} 値は，ソーストークン (ID_i) がターゲットトークン (ID_j) に高い注意を払うことを意味する．本章では，T^{ee} によって行われる Detokenization の例を示し (§ 3.1)，どのヘッドが Detokenization に実際に貢献しているかを調査する (§ 3.2)．

3.1 Detokenization の例

本節では，ソーストークンが続くと意味のある単語や句を形成するトークンに，注意ヘッドが高い T^{ee} 値を割り当てる Detokenization の例を示す．

GPT-2 トークナイザによって 2 つのトークンに分割される単語や句（例: “sapiens”）について，2 番目のトークン（“iens”）の ID を ID_i として固定し，すべてのヘッドのすべての $\mathrm{ID}_j \in V$ に対して T^{ee} を計算する．意味のある単語や句を形成し，高い T^{ee} を持つ ID_j と ID_i の組を，表 1 に示す．例えば，ID_i が “iens” に対応する場合，“_sap” と “Al” は，50,257 の語彙の中でヘッド#7 で最も高い T^{ee} スコアを与えられ，“_sapiens” と “Aliens” を Detokenize する[3)]．

表 1 Detokenization の例．ソーストークンが “iens” のとき，T^{ee} はターゲットトークンが “_sap” のときにヘッド #7 と#4 で最も大きくなる．

ヘッド	Token i	Token j (順位)	Detokenization
4	iens	_sap (1)	_sapiens
4	iens	_Sap (3)	_Sapiens
7	iens	_sap (1)	_sapiens
7	iens	Al (2)	Aliens
7	iens	_al (5)	_aliens
7	_Jackson	_Peter (1)	_Peter_Jackson
7	_Jackson	_Jesse (2)	_Jesse_Jackson
7	_Jackson	_Michael (3)	_Michael_Jackson

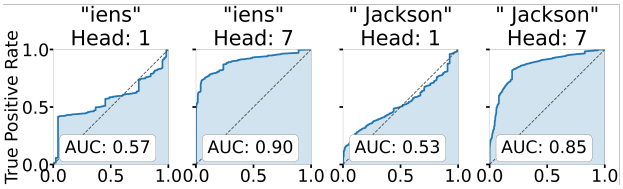

図 2 ソーストークンが “iens”, “ Jackson” のときの ROC 曲線．

3.2 Detokenization に貢献するヘッド

本節では，T^{ee} のスコアと Bigram の頻度の関係を調査することで，各注意ヘッドが Detokenization にどの程度貢献しているかを定量的に調査する．

固定された位置 i のトークンに対して，まず GPT-2 の語彙に含まれる 50,257 のトークンすべてに対して T^{ee} を計算する．次に OpenWebText Corpus [16] の Bigram カウントを使用して，T^{ee} がある閾値を超える Bigram カウントの割合を真陽性率として定義し，AUROC を計算する．図 2 では，位置 i のトークンが “iens” または “Jackson” に固定されたときのヘッド #7 とヘッド#1 の ROC 曲線を示す．高い AUC (Area Under Curve) は，高いスコアの T^{ee} が頻繁な Bigram を再構築する可能性が高いことを示している．更にすべての $\mathrm{ID}_i \in V$ について AUC を計算し，平均を取ることで，各注意ヘッドが Bigram を再構築するのにどの程度貢献しているかを定量化すると，表 2 の結果が得られる．この結果は，表 1, 3 でも観察されたヘッド#7 が最も大きな AUROC を持つことを示しており，Detokenization への寄与を裏付けている．

3) 付録 表 3 では，複数のトークンに分割される単語，人名や科学物質の Detokenization に寄与する他の T^{ee} の例を示す．

表 2　各ヘッドの平均 AUROC．AUROC が高いヘッドは Bigram の再構成に寄与することを示す．

Head	AUROC	Head	AUROC	Head	AUROC
7	0.88	4	0.69	8	0.44
11	0.81	3	0.63	1	0.40
6	0.79	10	0.62	9	0.40
0	0.73	2	0.55	5	0.29

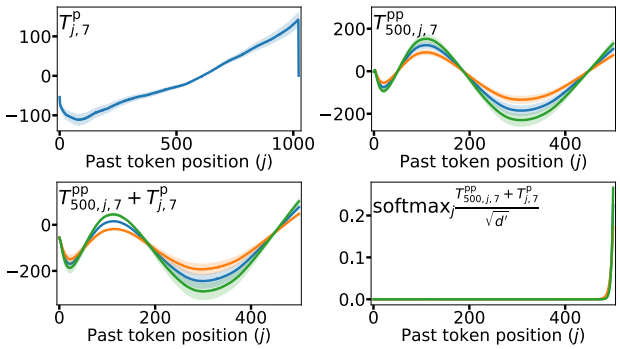

図 3　**左上**にヘッド#7 の T^{p}_j を示す．**右上**には現在トークンの位置 $i = 500$ のときの T^{pp} を示し，T^{p} との和を**左下**に示す．さらに Softmax 関数を通した結果を**右下**に示す．薄い色で塗られた部分は異なる e_{ID_i} による分散を示しており，青，緑，オレンジ の線はそれぞれ σ_j をその平均，最小，最大値で代表した時の結果を示している

4　Detokenization と位置情報

Detokenization を行うためには近い Token へ高い注意を払う必要がある．本章では，位置情報に基づく自己主張項 T^{p} と比較項 T^{pp} に焦点を当てる．

T^{p} は，位置情報に基づく自己主張項であり，図 3 (左上) より，過去のトークン位置 j に関して単調増加することがわかる．デコーダーモデルである GPT-2 は，過去のトークンに対してのみ注意を払うため，例えば $i = 200$ のとき，図 3 の $i = 200$ より右側の部分は無視される．T^{p} 項は j に関して単調増加しているため，近くのトークンに高い注意が割り当てられていることがわかる．

T^{pp} は，位置情報に基づく比較項であり，図 3 (右上) より，波打つようなパターンが観察される．また，T^{p} と同様に最も直近のトークンに高い注意が割り当てられていることがわかる．以上 2 つの項，T^{p} と T^{pp} の和をとり (図 3 左下)，実際の注意機構のように Softmax を取ると更に近いトークンへの注意は更に顕著になる．

5　経験的な確認

§ 4 では言語モデルの重みを分析し，近いトークンへの注意が高くなることを示したが，これが

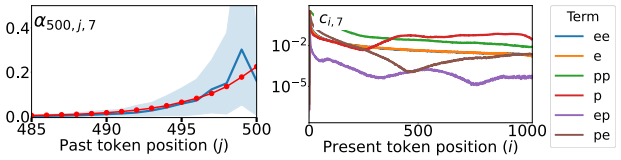

図 4　左に，OpenWebText Corpus の自然言語文を入力したときの，ヘッド#7，$i = 500$ のときの注意の重み $\alpha_{500,j,7}$ を青線で示す．赤線は図 3 の右下の図に対応する．右図は式 14 で計算される，各項の $\alpha_{i,j,7}$ への寄与の大きさを示す．

実際のモデル推論でも観察されるかを確認する．OpenWebText Corpus [16] を用いて，最初の Attention 層から実際の $\alpha_{i,j,h}$ を取得し，$i = 500$ のときの結果を図 4 の左図に示す．赤い線は位置情報に由来する項のみで分析した図 3 の右下図に対応し，青い線が実際の注意の重みを示している．赤い線と青い線は概ね同じ挙動をするが，トークン関連度も含んでいる青い線は直前のトークンに対して高い注意を向けていることがわかる．

更に，本稿では T^{ee} および T^{p}，T^{pp} のみを扱ったがそれらの項及び他の項が式 14 の中でどの程度重要であるかを調査する．各項の寄与を定量的に評価するために KL-Divergence を用いる．例えば，T^{ee} の寄与 c^{ee} は以下のように定義する:

$$\alpha'_{i,j,h} = \underset{\boldsymbol{x}_j \in \boldsymbol{X}, j \le i}{\mathrm{softmax}}\left((s_{i,j,h} - T^{\mathrm{ee}}_{i,j,h})/\sqrt{d'}\right) \tag{15}$$

$$Q_{i,h} = \begin{bmatrix} \alpha_{i,0,h} & \cdots & \alpha_{i,i,h} \end{bmatrix} \tag{16}$$

$$P_{i,h} = \begin{bmatrix} \alpha'_{i,0,h} & \cdots & \alpha'_{i,i,h} \end{bmatrix} \tag{17}$$

$$c^{\mathrm{ee}}_{i,h} = D_{\mathrm{KL}}(P_{i,h} || Q_{i,h}) \tag{18}$$

図 4 は，§§ 3, 4 で分析した 3 つの項が比較的高い寄与を持っていることを示している．

6　まとめ

我々は，Detokenization のメカニズムに迫るため，GPT-2 の最初の Attention 層を分解し，トークン埋込由来の計算と位置情報由来の計算を分離して分析した．トークン間の関連度に関しては，トークン埋め込みに由来する比較項が貢献していることを示した．また，位置情報由来の自己主張項や比較項は，相対的に近いトークンに高い注意を張ることを示した．これら 2 つの結果はどちらもモデルの重みを分析することで得られており，連続し，かつ関連性の高いトークンに大して高い注意を払うことで Detokenization に寄与していることを示した．

謝辞

本研究は，JST/CREST (JPMJCR20D2)，JST/博士後期課程学生支援 (JPMJBS2421) の支援を受けたものである．

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In **Advances in Neural Information Processing Systems**, Vol. 30, 2017.

[2] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and Others. Language models are unsupervised multitask learners. OpenAI blog, 2019.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Vol. 33, pp. 1877–1901, 2020.

[4] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Ahmad Kadian, Abhishek Al-Dahle, and others. The Llama 3 herd of models. **arXiv [cs.AI]**, 31 July 2024.

[5] Gemma Team. Gemma 2: Improving open language models at a practical size. **arXiv [cs.CL]**, 31 July 2024.

[6] Taku Kudo. Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. In **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 66–75, 2018.

[7] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural Machine Translation of Rare Words with Subword Units. In **Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1715–1725, 2016.

[8] Vedang Lad, Wes Gurnee, and Max Tegmark. The Remarkable Robustness of LLMs: Stages of Inference? In **ICML 2024 Workshop on Mechanistic Interpretability**, 24 June 2024.

[9] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. **arXiv [cs.LG]**, 2 May 2023.

[10] Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. From tokens to words: On the inner lexicon of LLMs. **arXiv [cs.CL]**, 8 October 2024.

[11] Ofir Press, Noah Smith, and Mike Lewis. Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation. In **International Conference on Learning Representations**, 29 January 2022.

[12] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with Rotary Position Embedding. **Neurocomputing**, Vol. 568, No. 127063, p. 127063, 1 February 2024.

[13] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 12216–12235. Association for Computational Linguistics, December 2023.

[14] Michael Hanna, Ollie Liu, and Alexandre Variengien. How does GPT-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. In **Thirty-seventh Conference on Neural Information Processing Systems**, 2 November 2023.

[15] Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards Automated Circuit Discovery for Mechanistic Interpretability. **Advances in Neural Information Processing Systems**, Vol. 36, pp. 16318–16352, 15 December 2023.

[16] Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. OpenWebText Corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

A 表記法

$$\boldsymbol{E} := \begin{bmatrix} \boldsymbol{e}_1 \\ \vdots \\ \boldsymbol{e}_{|V|} \end{bmatrix} \quad \in \mathbb{R}^{|V| \times d} \tag{19}$$

$$\boldsymbol{P} := \begin{bmatrix} \boldsymbol{p}_1 \\ \vdots \\ \boldsymbol{p}_L \end{bmatrix} \quad \in \mathbb{R}^{L \times d} \tag{20}$$

$$\boldsymbol{X} := \begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_n \end{bmatrix} \quad \in \mathbb{R}^{n \times d} \tag{21}$$

$$\boldsymbol{W}^O := \begin{bmatrix} \boldsymbol{W}_1^O \\ \vdots \\ \boldsymbol{W}_H^O \end{bmatrix} \quad \in \mathbb{R}^{d \times d} \tag{22}$$

$$\boldsymbol{W}^Q := \begin{bmatrix} \boldsymbol{W}_1^Q & \cdots & \boldsymbol{W}_H^Q \end{bmatrix} \quad \in \mathbb{R}^{d \times d} \tag{23}$$

$$\boldsymbol{W}^K := \begin{bmatrix} \boldsymbol{W}_1^K & \cdots & \boldsymbol{W}_H^K \end{bmatrix} \quad \in \mathbb{R}^{d \times d} \tag{24}$$

$$\boldsymbol{W}^V := \begin{bmatrix} \boldsymbol{W}_1^V & \cdots & \boldsymbol{W}_H^V \end{bmatrix} \quad \in \mathbb{R}^{d \times d} \tag{25}$$

$$\boldsymbol{b}^Q := \begin{bmatrix} \boldsymbol{b}_1^Q & \cdots & \boldsymbol{b}_H^Q \end{bmatrix} \quad \in \mathbb{R}^{d} \tag{26}$$

$$\boldsymbol{b}^K := \begin{bmatrix} \boldsymbol{b}_1^K & \cdots & \boldsymbol{b}_H^K \end{bmatrix} \quad \in \mathbb{R}^{d} \tag{27}$$

$$\boldsymbol{b}^V := \begin{bmatrix} \boldsymbol{b}_1^V & \cdots & \boldsymbol{b}_H^V \end{bmatrix} \quad \in \mathbb{R}^{d} \tag{28}$$

$$\boldsymbol{I} := \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \quad \in \mathbb{R}^{d \times d} \tag{29}$$

$$\boldsymbol{1} := \begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \quad \in \mathbb{R}^{d} \tag{30}$$

B Detokenization の例

表 3 表 1 以外の Detokenization の例.

ヘッド	Token i	Token j (順位)	Detokenization
4	tarian	Liber (1)	Libertarian
4	tarian	_Liber (2)	_Libertarian
3	yo	Tok (1)	Tokyo
3	yo	_Tok (2)	_Tokyo
4	yo	Tok (6)	Tokyo
7	yo	Tok (1)	Tokyo
4	ation	anim (1)	animation
4	ation	_Represent (3)	_Representation
7	ation	_dict (1)	_dictation
7	ation	_Fabric (3)	_Fabrication
7	ation	_coron (5)	_coronation
7	ation	valid (7)	validation
7	ation	Gener (8)	Generation
4	_Korea	_North (1)	North_Korea
7	_Korea	_North (1)	_North_Korea
7	_Korea	North (2)	North_Korea
7	_Korea	_South (3)	_South_Korea
7	_Korea	South (4)	_South_Korea
1	_Obama	_Barack (3)	_Barack_Obama
1	_Obama	President (8)	_President_Obama
4	_Obama	_Barack (3)	_Barack_Obama
5	_Obama	_Barack (5)	_Barack_Obama
7	_Obama	_President (2)	_President_Obama
7	_Obama	_Michelle (3)	_Michelle_Obama
7	_Einstein	_Albert (1)	_Albert_Einstein
7	_Einstein	Albert (2)	Albert_Einstein
7	_Jackson	_Michael (1)	_Michael_Jackson
7	_Jackson	_Peter (2)	_Peter_Jackson
7	_Jackson	Michael (3)	Michael_Jackson
7	_Jackson	_Jesse (4)	_Jesse_Jackson
7	_Jackson	Peter (5)	Peter_Jackson
7	_Jackson	_Janet (6)	_Janet_Jackson
7	_chloride	_aluminum (1)	_aluminum_chloride
7	_chloride	_copper (3)	_copper_chloride
7	_chloride	_vinyl (6)	_vinyl_chloride
7	_chloride	_sodium (7)	_sodium_chloride
7	_chloride	_platinum (8)	_platinum_chloride
10	_chloride	_potassium (2)	_potassium_chloride
10	_chloride	_sodium (3)	_sodium_chloride
3	_century	_19 (1)	_19_century
3	_century	_nineteenth (7)	_nineteenth_century
7	_century	_21 (1)	_21_century
7	_century	_twentieth (6)	_nineteenth_century