Tighter Locality：大規模言語モデルにおける知識編集手法のより厳密な局所性評価

片岡晴彦　辻航平　坂井優介　上垣外英剛　渡辺太郎
奈良先端科学技術大学院大学
{kataoka.haruhiko.kl7, tsuji.kohei.tl1}@naist.ac.jp
{taro,kamigaito.h,sakai.yusuke.sr9}@is.naist.jp

概要

大規模言語モデル（LLM）の知識を効率的に変更するための知識編集手法は、ユーザの意図に合わせた柔軟な編集を行える必要がある．そのためにはどの知識を更新・保存するかを制御できる必要があり，無関係な知識への影響力を測る指標である局所性の評価は重要である．しかし既存ベンチマークの局所性評価用データには，編集知識と主語・目的語を共有する別の知識が不足しておりこの能力の評価が不十分である．本研究ではこのような知識を収集し構築した新たなベンチマーク Tighter Locality を提案する．実験の結果，既存の知識編集手法は編集知識と主語・目的語を共有する別の知識に対する影響力の制御に課題があることがわかった．

1　はじめに

大規模言語モデル（LLM）の知識を，局所的かつ効率的に更新するための知識編集に関する研究が注目を集めている．知識編集は，主に外部記憶ベースの手法 [1, 2]，全体最適化ベースの手法 [3, 4, 5]，局所変更ベースの手法 [6, 7, 8, 9, 10] に大別される．外部メモリベースの手法は，編集された知識を知識グラフなどの外部メモリに蓄積し，その知識に関わる推論毎にメモリを参照する手法である．全体最適化ベースの手法は，Fine-tuning によるモデルパラメータの大部分の編集などを含む手法である．局所編集ベースの手法は，モデル内部に，ある特定の知識をエンコードしているニューロンの集合があると仮定し，その部分を特定・編集する手法である．

局所編集ベースの手法は，モデルが学習したパラメータと膨大な知識との関係性を解明しようとする試みである．モデルパラメータの解釈については多くの研究がされており，過去にはモデルの

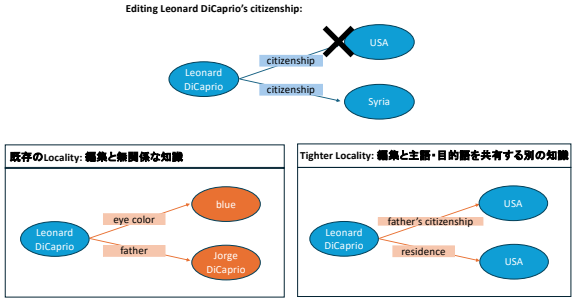

図 1　：既存ベンチマークの Locality 評価用データには編集知識と主語・目的語を共有する別の知識はほとんど含まれていなかった．Tighter Locality はそのような知識を収集した．

Feed-Forward 層がキー・バリューメモリとして機能しているという可能性 [11] が示された．知識編集の文脈ではこの観察を受けて，Integrated Gradients [12] を用いた，特定の知識をエンコードするニューロン，いわゆる「知識ニューロン」の概念の提案 [9] もあった．局所編集ベースの手法である ROME [6] や MEND [8] は一定程度の成功を収めている．

また，知識編集手法の評価も多様なベンチマークの提案とともに多角的に行われている [2, 13, 14, 15, 16]．特に，編集対象の知識と無関係な知識に対して悪影響を与えていないかを測る指標である局所性の評価は，ユーザの意図に合わせた柔軟な編集を実現するために重要であるが，我々はこの評価が不十分であると考える．既存の局所性評価では，編集対象と無関係な知識が保持されているかどうかが主に検証されてきた．しかし実際の応用においては，編集対象と主語や目的語を共有する関係的に近接した知識への影響をどのように制御できるかが重要となる．例えば，"Leonardo DiCaprio" の "citizenship" を "Syria" に編集する場合，"eye color" のような明らかに無関係な属性が変化してはならな

いことは自明である一方で，"Leonard DiCaprio" の "father's citizenship" を同時に変更すべきか否かは，ユーザの意図に依存する図 1．しかし既存の局所性評価用ベンチマークには，このような主語・目的語を共有する関連知識がほとんど含まれておらず，ユーザの意図に応じた細粒度な局所性制御を評価することが困難である．

したがって，本研究では編集知識と主語・目的語を共有する別の知識により構成された局所性評価用データを含むベンチマーク Tighter Locality の提案を行う．実験の結果，既存の知識編集手法は編集知識と主語・目的語を共有する別の知識への影響が強く，意図しない知識まで変更してしまう傾向が強いことがわかった．

2 LLMs の知識編集

知識編集 本研究では過去の研究にならい，主語 s，目的語 o とその間の関係 r を用いて，ある知識を (s,r,o) で定義する．その上で，o 以外の目的語 o' をとるようなある知識編集 e を $e:(s,r,o)\rightarrow o'$ と表記する．また，あるパラメータ θ を持つモデル f_θ に対し，入力 x の次トークン予測位置において語彙 $\mathbb{V}$ 上の各トークン v に割り当てられる確率を $p_\theta(v\mid x)$ とする．この時，ある編集対象のプロンプト x_e に対して知識編集を

$$\arg\max_{v\in\mathbb{V}} p_{\theta'}(v\mid x_e)=o'$$

を満たすようなパラメータの編集 $\theta\rightarrow\theta'$ と定義する．

局所性（Locality） 知識編集は x_e に対するモデルの挙動を変化させつつ，無関係な入力に対する出力を保存する必要がある．ここで任意の入力集合を X，編集 e に伴い出力の変更が望まれるような入力の集合を X_e とする．この時，ある編集 e における局所性を以下のような制約として定義する:

$$\arg\max_{v\in\mathbb{V}} p_\theta(v\mid x)=\arg\max_{v\in\mathbb{V}} p_{\theta'}(v\mid x), x\in X\setminus X_e$$

評価指標 本研究では各知識編集手法を主に **Accuracy**, **Portability**, **Locality** の三つの指標で評価する．Accuracy では編集の成功率を評価する．また，知識編集は，単に特定のプロンプトに対する LLM の出力を変更するだけでなく，編集に伴って更新されるべき関連知識を適切に反映しつつ，無関係な知識への影響を最小限に抑える能力が求められる．この能力をそれぞれ Portability, Locality で評価する．本研究でのスコアリングについては 付録 A に記載している．

知識編集手法 本研究では既存の 3 つの知識編集手法を用いて実験を行う．

ROME [6] は LLM の重みに着目した知識編集手法であり LLM の数層の FFN のみを編集する．

MEND [8] は局所編集ベースに分類されるが，メタ学習的アプローチとしても位置付けられる．編集対象の入力と損失から得られる勾配を入力として，重み更新量を直接出力するメタネットワークを学習する．これにより，単一の前向き計算によって局所的な重み更新を生成し，所望の知識編集を実現する．

SWEA [10] は LLM の埋め込みベクトルに着目した局所編集手法である．具体的には編集対象の知識に対応する主語トークンの埋め込み表現に対して差分ベクトルを学習し，推論時に当該埋め込みを修正することで知識編集を実現する．

3 Tighter Locality

より厳密な局所性 既存ベンチマークの局所性評価用データは編集知識と主語・目的語を共有しないような知識が大部分を占める．編集知識と無関係な知識への影響を評価するために，既存のベンチマークの局所性評価用データは依然として重要である一方で，ユーザの意図に沿った柔軟な知識編集の実現性を評価するには不十分である．したがって本研究では，ある知識編集 $(s,r,o)\rightarrow o'$ に関して，より厳密な局所性評価用データとして (s,r',o) を定義する．例えば編集 $(\text{Leonard DiCaprio}, \text{citizenship}, \text{USA})\rightarrow \text{Syria}$ に対して，$(\text{Leonard DiCaprio}, \text{father's citizenship}, \text{USA})$ などが定義されることになる．

データの収集 データセットは知識編集フレームワークである EasyEdit [16, 17, 18, 19, 20, 21] 上で公開されている Wiki_Counterfact データセットを拡張する形で，wikidata [22] から収集する．具体的には編集知識 (s,r,o) に対して 1-hop, 2-hop の Tighter Locality 知識を以下の手順で収集する．

まず，Wikidata 上で主語 s と接続する全ての関係の集合 M_s を取得する:

$$M_s:=\{r\mid(s,r,o')\text{is a fact}\}$$

また M_s から編集対象の関係 r を除いた関係の集合

を $M_s^{\setminus r}$ とする．$M_s^{\setminus r}$ 上の任意の関係を介して目的語 o と接続する全ての関係 $r_{s,o}$ を取得し，1-hop の Tighter Locality 知識を定義する:

$$r_{s,o} \in \{r' \in M_s^{\setminus r} | (s, r', o)\}$$

$$\text{1-hop Tighter Locality} =: (s, r_{s,o}, o)$$

次に $\mathbb{O}$ を全ての目的語の集合とし， s と，o 以外の任意の目的語 $\bar{o} \in \mathbb{O}^{\setminus o}$ を接続する全ての関係 $r_{s,\bar{o}}$ を取得する:

$$r_{s,\bar{o}} \in \{r' \in M_s^{\setminus r} | (s, r', \bar{o})\}$$

$\bar{o}$ を新たな主語として， $\bar{o}$ と接続する全ての関係 $M_{\bar{o}}$ 中で，目的語 o と接続する関係を $r_{\bar{o},o}$ とする．

$$M_{\bar{o}} := \{r | (\bar{o}, r, o') \text{is a fact}\}$$

$$r_{\bar{o},o} \in \{r'' \in M_{\bar{o}} | (\bar{o}, r'', o)\}$$

この $r_{\bar{o},o}$ を用いて 2-hop の Tighter Locality 知識を定義する:

$$\text{2-hop Tighter Locality} := (s, r_{s,\bar{o}} \circ r_{\bar{o},o}, o)$$

ただし，$\circ$ は関係の合成を意味し，例えば "father" $\circ$ "citizenship" は "father's citizenship" などとなる．

データセットは編集用プロンプト，Portability 評価用プロンプト，Locality 評価用プロンプト，Tighter_Locality 評価用プロンプトを含む編集エンティティ 826 件から構成される．Portability 用プロンプトが 5135 件，Locality 用プロンプトが 5388 件，Tighter_Locality 用プロンプトが 3753 件である．

4 実験

実験設定 実験は Llama-3.1 8B [23], Qwen2.5 7B [24, 25], GPT-J 6B [26], GPT2 XL [27] で行う．また ROME, MEND での実験は知識編集フレームワーク EasyEdit を利用する．評価方法について，**Accuracy** は，編集対象のプロンプト直後の予測位置において，編集目標の目的語 o' がロジット値で最大となった割合で評価する．**Portability** は，Portability 評価用プロンプトに対して正解トークンが誤りトークンをロジット値で上回った割合 (**ToO**: TargetOver-Original) で評価する．**Locality** は Wiki_Counterfact データセットから引き継いだ Locality プロンプトに対して各モデルの編集前後で出力した 20 トークンのテキストの意味ベクトルのコサイン類似度で評価する．**Tighter Locality** はコサイン類似度，ToO の 2 つの指標で評価する．

表 1 Tighter Locality 上での知識編集実験結果
ボールドは各モデル・指標での最高スコア．

	Editor	Acc	Port	Loc(cos)	TLoc	TLoc(cos)
GPT-J-6B	ROME	**98.7**	**62.8**	29.3	5.91	34.4
	MEND	60.7	24.1	**83.0**	**49.0**	**81.5**
	SWEA	85.1	61.6	31.5	22.1	41.5
Llama3.1-8B	ROME	**97.2**	**64.5**	42.6	12.2	43.0
	MEND	76.5	39.6	**73.8**	**61.3**	**74.5**
	SWEA	72.9	63.3	35.4	31.0	41.0
Qwen2.5-7B	ROME	**97.6**	**62.4**	40.7	20.5	44.9
	MEND	68.0	22.8	**84.2**	34.5	**85.7**
	SWEA	63.5	59.4	35.9	**38.5**	41.9
GPT2-XL	ROME	**94.2**	60.1	33.2	22.9	41.0
	MEND	52.8	24.7	**73.6**	**41.2**	76.0
	SWEA	76.5	**65.5**	32.8	27.7	**83.0**

結果 Tighter Locality 上での実験結果を表 1 に示す．ROME は Accuracy, Portability では高スコアを得る傾向がある一方で，Locality 系指標のスコアが一貫して低い．特に Tighter Locality の ToO ではどのモデルに対しても最もスコアが低い．これは編集 $(s, r, o) \rightarrow o'$ が，編集知識と主語・目的語を共有する別の知識に対しても大きな影響を与えていることを示している．MEND は Locality 系指標，特にコサイン類似度でのスコアが比較的高いが，編集成功率が低いことに一部起因すると考えられる．SWEA は Portability のスコアが ROME に次いで高く，GPT2 では最も高スコアである．また SWEA と ROME の Tighter Locality での結果を比べると，ToO では一貫して SWEA が上回る．これは SWEA が編集する位置が主語トークンの埋め込みであり，編集の影響を編集位置以降の層で緩和させる余地が ROME よりも大きいことが関係しているかもしれない．また，Locality, Tighter Locality におけるコサイン類似度での評価には大きな違いが見られない．この結果は既存の Locality と Tighter Locality のいずれに対しても編集が影響を及ぼしていることを示している．一方で，既存の Locality では編集の影響がどの目的語に帰属するのかを事前に特定できないため誤りトークン集合を定義することが難しく，Perplexity に基づく定量評価が困難であるという問題があった．Tighter Locality では影響が帰属する目的語が明確に予測できることから，解釈性の高い評価が可能である．

5 考察

定性分析 表 2 に各編集手法による編集前後の Llama 3.1 8B による推論例を示す．編集前のモデルの推論結果が現実世界の知識と一致しない例も一

表 2　各種法での編集前後のモデルでの Locality,Tighter Locality に対する推論例

Edit	query type	query	Pre	SWEA	ROME	MEND
(Leonard DiCaprio, citizenship, USA) → Syria	Loc	(Leonard DiCaprio, eye color)	brown	brown	brown	Syria
	Tloc(ours)	(Leonard DiCaprio, place of birth)	USA	Syria	Syria	Sydney
(Jerrod Carmichael, citizenship, USA) → Terenganu	Loc	(Jerrod Carmichael, gender)	male	male	Terenganu	Male
	Tloc(ours)	(Jerrod Carmichael, country ∘ organization)	USA	Terenganu	Terenganu	Carmichaelation
(Tony Curtis, citizenship, USA) → British Leeward Islands	Loc	(Tony Curtis, gender)	male	male	male	Male
	Tloc(ours)	(Tony Curtis, child+citizenship)	USA	Leeward Islands	British Leeward Islands	British
(Clint Eastwood, citizenship, USA) → liberal party	Loc	(Clint Eastwood, gender)	male	male	liberal conservation	male
	Tloc(ours)	(Clint Eastwood, ethnic group ∘ country)	USA	liberal party	liberal party	liberal party
(Ranbir Kapoor, place of birth, Mumbai) → Blackburg	Loc	(Ranbir Kapoor, gender, male)	male	male	male	male
	Tloc(ours)	(Ranbir Kapoor, father ∘ place of birth)	Munbai	not known	Blackburg	BlackshownAmardeep
(Fatima Sana Shaikh, citizenship, India) → Saxony	Loc	(Fatima Sana Shaikh, gender)	female	female	female	female
	Tloc(ours)	(Fatima Sana Shaikh, educated)	India	Saxony	Saxony	Saxena Vidya Mandir

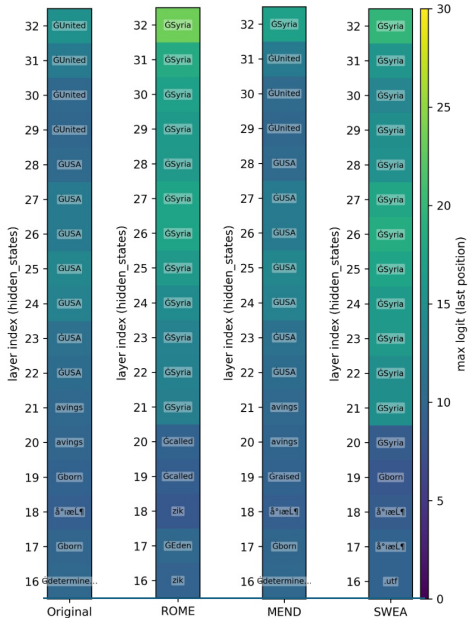

図 2　各手法での編集前後における logit lens の結果
入力プロンプト末尾での状態を取得．図は編集 (Leonard DiCaprio, citizenship, USA) → Syria 前後での Tighter Locality (Leonard DiCaprio, place of birth) に対するもの．

部含むが，編集の影響を評価することが目的であるため実際の知識との整合性を必ずしも重要視しない．"Leonard DiCaprio" の例では，既存の Locality プロンプトに対してのモデルの推論結果は，MEND で編集したもの以外影響が見られない．一方で Tighter Locality プロンプトに対しての推論結果は全てのモデルで影響が見られる．特に SWEA, ROME での編集結果はともに "Syria" となっており，これは編集知識に対しての挙動の変化と一致している．他の例でも同様の傾向が見られることから，既存の知識編集手法は編集知識と主語・目的語を共有する別の知識に対する影響力の制御に課題があることがわかる．

logit lens　この節では，このような例に対する編集前後のモデルの内部状態の変化を観察するために，logit lens を用いた分析を行う．

図 2 に Llama 3.1 8B での結果の例を示す．例は編集 (Leonard DiCaprio, citizenship, USA) → Syria 前後のモデルによる Tighter Locality (Leonard DiCaprio, place of birth) に対する結果である．モデルの中間層を編集する ROME や埋め込みベクトルに介入する SWEA で編集されたモデルでは，誤りトークンである "Syria" が 20 層から最大 logit をとっている．また編集前モデルでは正解トークン "USA" が 22 層で最大 logit となる．回答するトークンが強調される層はほぼ一致していることから，編集はモデルの自然な挙動を保っていると考えられる．しかし同時に，編集知識と Tighter Locality 知識を区別していないことも読み取れる．付録 B に同編集における Accuracy, Locality プロンプトでの logit lens の結果を掲載する．

6　終わりに

本研究では，知識編集手法の新たなベンチマーク Tighter Locality の提案を行った．Tighter Locality は，編集知識と主語・目的語を共有する別の知識から構成された新たな局所性評価用データを含む．Tighter Locality 上での実験の結果，既存の知識編集手法は編集知識と主語・目的語を共有する知識に対する影響力の制御に課題があることがわかった．

ユーザの意図に沿った知識編集を実現するためより厳密な局所性の評価は重要である．本研究ではより厳密な局所性への対応という新たな課題を見出したが，更なる分析は今後の研究に委ねる．

参考文献

[1] Shikhar Murty, Christopher Manning, Scott Lundberg, and Marco Tulio Ribeiro. Fixing model bugs with natural language patches. pp. 11600–11613, December 2022.

[2] Zexuan Zhong, Zhengxuan Wu, Christopher Manning, Christopher Potts, and Danqi Chen. MQuAKE: Assessing knowledge editing in language models via multi-hop questions. pp. 15686–15702, December 2023.

[3] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP2021)**, 2021.

[4] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and learn: Fine-tuning deep pretrained language models with less forgetting. pp. 7870–7881, November 2020.

[5] Anton Sinitsin, Vsevolod Plokhotnyuk, Sergei Popov, and Artem Babenko. Editable neural networks. 04 2020.

[6] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. Vol. 35, pp. 17359–17372, 2022.

[7] Daniel Tamayo, Aitor Gonzalez-Agirre, Javier Hernando, and Marta Villegas. Mass-editing memory with attention in transformers: A cross-lingual exploration of knowledge. pp. 5831–5847, August 2024.

[8] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. 2022.

[9] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. pp. 8493–8502, May 2022.

[10] Xiaopeng Li, Shasha Li, Shezheng Song, Huijun Liu, Bin Ji, Xi Wang, Jun Ma, Jie Yu, Xiaodong Liu, Jing Wang, and Weimin Zhang. Swea: Updating factual knowledge in large language models via subject word embedding altering. **Proceedings of the AAAI Conference on Artificial Intelligence**, Vol. 39, pp. 24494–24502, 04 2025.

[11] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. pp. 5484–5495, November 2021.

[12] Qiqi Yan Mukund Sundararajan, Ankur Taly. Axiomatic attribution for deep networks. pp. 3319–3328, August 2017.

[13] Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. Can we edit factual knowledge by in-context learning? pp. 4862–4876, December 2023.

[14] Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. **Transactions of the Association for Computational Linguistics**, Vol. 12, pp. 283–298, 2024.

[15] Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. pp. 15202–15232, August 2024.

[16] Ningyu Zhang, Yunzhi Yao, Bo Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models. **ArXiv**, Vol. abs/2401.01286, , 2024.

[17] Peng Wang, Ningyu Zhang, Bozhong Tian, Zekun Xi, Yunzhi Yao, Ziwen Xu, Mengru Wang, Shengyu Mao, Xiaohan Wang, Siyuan Cheng, Kangwei Liu, Yuansheng Ni, Guozhou Zheng, and Huajun Chen. EasyEdit: An easy-to-use knowledge editing framework for large language models. pp. 82–93, August 2024.

[18] Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. pp. 10222–10240, December 2023.

[19] Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. Can we edit multimodal large language models? pp. 13877–13888, December 2023.

[20] Shengyu Mao, Xiaohan Wang, Mengru Wang, Yong Jiang, Pengjun Xie, Fei Huang, and Ningyu Zhang. Editing personality for large language models. pp. 241–254, 2024.

[21] Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. pp. 53764–53797, 2024.

[22] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. **Communications of the ACM**, Vol. 57, No. 10, pp. 78–85, 2014.

[23] Meta AI. The llama 3 herd of models. **arXiv preprint arXiv:2407.21783**, 2024.

[24] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, and additional authors. Qwen2 technical report. **arXiv preprint arXiv:2407.10671**, 2024.

[25] Qwen Team. Qwen2.5: A party of foundation models. September 2024.

[26] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. May 2021.

[27] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. **OpenAI blog**, Vol. 1, No. 8, p. 9, 2019.

A　評価指標の定義

モデル f_θ に対し，入力 x の次トークン予測位置において語彙 $\mathbb{V}$ 上の各トークン v に割り当てられる確率を $p_\theta(v \mid x)$，その softmax 前の値をロジット $\ell_\theta(v \mid x)$ と定義する:

$$p_\theta(v \mid x) = \frac{\exp(\ell_\theta(v \mid x))}{\sum_{v' \in \mathbb{V}} \exp(\ell_\theta(v' \mid x))}$$

編集集合を $\mathscr{E} := \{e_1, e_2, \ldots, e_N\}$ とし，各編集を $e_i : (s_i, r_i, o_i) \to o_i'$ と定義する．編集 e_i は (s_i, r_i) を表現するプロンプト x_{e_i} を用いて行われ，編集後のモデルを $f_{\theta^{(i)}}$ と表す．

Accuracy は，編集対象プロンプト x_{e_i} の直後の予測位置において，編集目標の目的語 o_i' が最も高い確率を持つ割合として定義する:

$$\textbf{Accuracy} := \mathbb{E}_{e_i \in \mathscr{E}} \mathbb{1}\left\{\arg\max_{v \in \mathbb{V}} p_{\theta^{(i)}}(v \mid x_{e_i}) = o_i'\right\}$$

Portability は，Portability 評価用プロンプト集合 $\mathscr{P}_i^{\text{port}}$ に対して，正解トークン o が誤りトークン $\tilde{o}$ をロジット値で上回る割合（ToO:Target-over-Original）として定義する:

$$\textbf{Portability} := \mathbb{E}_{e_i \in \mathscr{E}} \mathbb{E}_{x \in \mathscr{P}_i^{\text{port}}} \mathbb{1}\left\{\ell_{\theta^{(i)}}(o \mid x) > \ell_{\theta^{(i)}}(\tilde{o} \mid x)\right\}$$

Locality は，Wiki_Counterfact データセットに基づく Locality プロンプト集合 $\mathscr{P}_i^{\text{loc}}$ に対して，編集前後で生成された 20 トークンの出力列 $y^{\text{pre}}(x)$ と $y^{\text{post}}(x)$ の意味ベクトルのコサイン類似度により評価する．文埋め込み関数を $\phi(\cdot)$ とすると，

$$\textbf{Locality} := \mathbb{E}_{e_i \in \mathscr{E}} \mathbb{E}_{x \in \mathscr{P}_i^{\text{loc}}} \cos\Big(\phi\big(y^{\text{pre}}(x)\big), \phi\big(y^{\text{post}}(x)\big)\Big)$$

Tighter Locality は，編集知識と主語・目的語または関係を共有する Tighter Locality プロンプト集合 $\mathscr{P}_i^{\text{tight}}$ に対して評価を行う．具体的には，コサイン類似度と ToO の 2 つの指標を用いる:

$$\textbf{TLocality}_{\cos} := \mathbb{E}_{e_i \in \mathscr{E}} \mathbb{E}_{x \in \mathscr{P}_i^{\text{tight}}} \cos\Big(\phi\big(y^{\text{pre}}(x)\big), \phi\big(y^{\text{post}}(x)\big)\Big)$$

$$\textbf{TLocality}_{\text{ToO}} := \mathbb{E}_{e_i \in \mathscr{E}} \mathbb{E}_{x \in \mathscr{P}_i^{\text{tight}}} \mathbb{1}\left\{\ell_{\theta^{(i)}}(o \mid x) > \ell_{\theta^{(i)}}(\tilde{o} \mid x)\right\}$$

また本研究での文埋め込みは sentence-transformers/all-mpnet-base-v2 によって計算する．

B　logitlens の結果例

図 3，図 4 に，それぞれ Accuracy 評価用プロンプト，Locality 評価用プロンプトに対する推論時の logit lens の結果を示す．

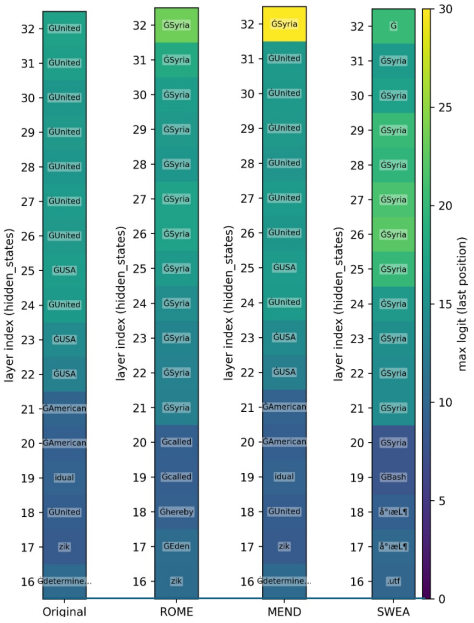

図 3　Accuracy プロンプト (Leonard DiCaprio, citizenship) に対する推論時の内部状態.

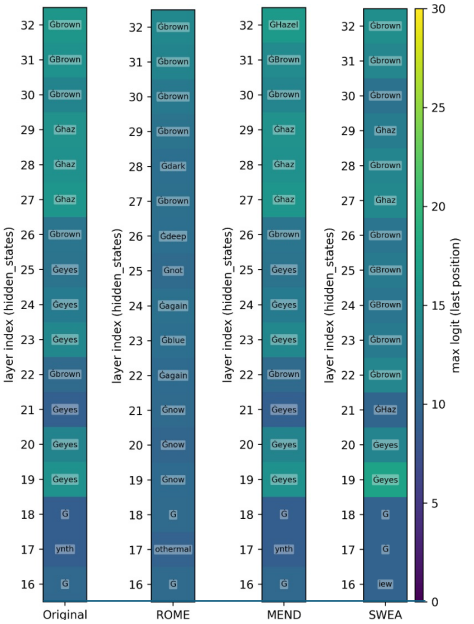

図 4　Tighter Locality プロンプト (Leonard DiCaprio, eye color) に対する推論時の内部状態.