言語モデルからの知識削除：頻出実体の知識は副作用が破滅的

高橋 良允 [1]　鴨田 豪 [1]　Benjamin Heinzerling[2,1]　坂口 慶祐 [1,2]　乾 健太郎 [3,1,2]
[1] 東北大学　[2] 理化学研究所　[3]MBZUAI
{ryosuke.takahashi, go.kamoda}@dc.tohoku.ac.jp
benjamin.heinzerling@riken.jp　keisuke.sakaguchi@tohoku.ac.jp
kentaro.inui@mbzuai.ac.ae

概要

言語モデル（LM）は学習を通じて内部に知識を獲得できるが，学習データに含まれる個人情報や機密情報によりプライバシーに関する課題が生じる．そのため，LM の知識削除に関する研究が活発に行われているが，LM に学習させる知識の構造を考慮した分析は十分に行われていない．本研究では，知識構造に着目して LM に対する知識削除の影響を分析し，頻出実体に関する知識の削除は大きな副作用を伴い破滅的であることを明らかにした．また，制御された知識構造で学習したモデルを使用して知識削除を分析した初めての研究であり，人工的な知識グラフを用いた知識削除に関する分析の新たな方向性を示した．

1　はじめに

言語モデル（LM）は学習によって内部のパラメータに知識を格納できることがわかっており，LM 内の知識を分析する研究が注目を集めている [1–3]．課題の一つに，LM は学習データが収集された時点までの知識しか持っておらず，絶えず変化する実世界の知識に対して頑健でないということが挙げられる [4]．また，LM は学習データに含まれる個人情報や機密情報を漏洩してしまう可能性があり，プライバシーに関する課題も存在する [5]．これらの課題に対処するために，LM の**知識編集** [6–11] や，**知識削除** [12, 13] に関する研究が積極的に行われている．これらの研究では知識の編集や削除に関して一部の成功事例が報告される一方で，失敗事例や課題，困難が存在することが明らかになっている．

本研究の目標は，**知識の編集や削除が意図通りに機能する条件とその理由を理解する**ことであり，削除される知識の種類に着目することが重要であるという仮説のもと分析を行う．本稿では，特に知識に関連する実体の，学習コーパスにおける頻度に着目する (図図 1)．この仮説を検証するために，本研究では制御された実験を設計し，知識削除が LM に与える影響を分析する．我々は「知識の種類」という概念を，LM が学習する知識グラフの構造的特性にて形式化する．異なる特性を持つ人工的な知識グラフを作成し，これらのグラフ上で LM を訓練することで学習される知識を正確に操作する．得られたモデルに対して特定の実体に関する知識削除を行い，削除による影響を分析する．

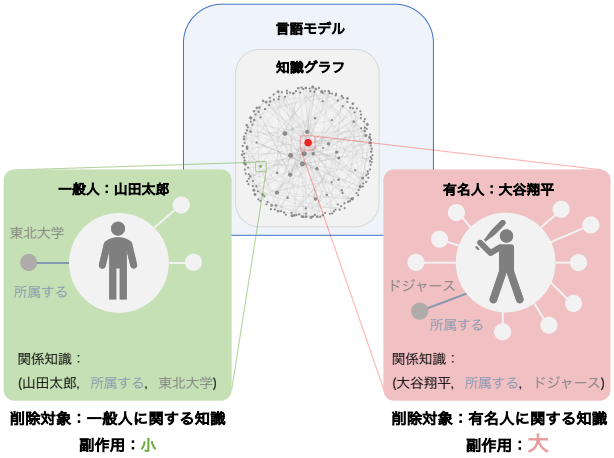

図 1　知識グラフは実体どうしの関係知識を表現する．例えば，“山田太郎が東北大学に所属している” という知識は（山田太郎，所属する，東北大学）のように表現される．それぞれの実体は次数 (他の実体と関連する数) を持ち，例えば有名人の次数は大きく，一般人の次数は小さいと考えられる．言語モデルは，削除対象の知識に関わる実体の次数が大きいほど副作用が破滅的である．

その結果，実世界の知識構造を持つ LM において，**頻出実体に関する知識の削除による副作用は大きいため破滅的であり，低頻度実体に関する知識は副作用が小さい**ことを明らかにした．また，本研究は，知識構造を制御したモデルで知識削除の分析を行った初めての研究であり，**人工知識グラフを用いた知識削除に関する分析の新たな方向性**を示した．

2　実験の設計・方針

LM において，知識は複数のモデルパラメータに分散してエンコードされ，知識間で複雑な相互依存関係を構築しており，特定の知識を削除することで予期せぬ**副作用**を引き起こす可能性がある．また，すべての知識の重要度は等価ではなく，多くの他の知識と関連する**重要な知識**や関わりが少ない**重要でない知識**のような，知識の種類が存在すると考えられる．しかし，LM が事前学習によって内部にどのような知識構造を獲得しているかを正確に知ることは困難であり [2]，既存の事前学習済み LM を用いて知識構造を考慮した詳細な分析を行うことは難しい．そこで，本研究では次のような実験方針を採る．まず人工的な知識グラフを作成し (3.1 項)，LM に学習させる（3.2 項）．これにより **LM が学習によって獲得する知識構造を正確にコントロールする**．そして，知識が制御された LM に対して既存の知識削除手法を用いて知識の削除を行い（3.3 項，3.4 項），仮説検証のための分析を行う（4 節）．本研究は，このような実験設計のもとで，知識構造に着目した知識削除に関する分析を実現する．

3　実験設定

3.1　知識グラフ

本研究では，(山田太郎, 所属する, 東北大) のような (s, r, o) [1] の三組で表される関係知識を扱う．扱う関係知識を，s と o が頂点に，r が辺に対応するように表現したものを知識グラフと呼ぶ．ここで，s，o は実体の集合 $\mathbb{E}$ の，r は関係の集合 $\mathbb{R}$ の元である [2]．

これまでの知識編集・削除に関する研究では，上記の例のような自然言語で表現された実体どうしの関係知識を表現した知識グラフを前提としていた．本研究では，LM が扱う知識をコントロールするために，人工的な知識グラフを作成する．今回実験に用いるグラフの構造は，頂点間に辺を張る確率が一様となるように構成される Erdős-Rényi（ER）グラフ [14] と，頂点の次数がべき乗則に従い，より実世界の設定に近いとされる Barabási-Albert（BA）グラフ [15] の 2 種類である（図 2）．BA グラフでは極端に次数の大きい実体や小さい実体を作成することができ，そのような実体に対して知識削除の分析が可能となることが人工知識グラフの利点である．

1）　s: subject，o: object，r: relation
2）　後述の人工知識グラフは，$|\mathbb{E}| = 200$，$|\mathbb{R}| = 50$ となるように作成する．

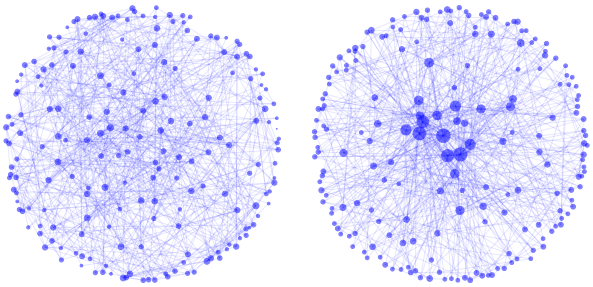

図 2　（左）Erdős-Rényi グラフ．頂点の次数に偏りが無く，シンプルな構造．（右）Barabási-Albert グラフ．頂点の次数に偏りがあり，実世界に近い構造．

表 1　異なる層数の GPT モデルにおける，知識グラフ学習後の訓練データと全データにおける正解率

知識グラフ	モデル	正解率：訓練データ	正解率：全データ
ER	6 層	.9998	.9941
	12 層	.9985	.9983
	24 層	.9900	.9905
BA	6 層	.9991	.9987
	12 層	.9991	.9989
	24 層	.9972	.9964

3.2　LM の知識

作成した知識グラフのもとで知識削除の分析を行うため，LM の事前学習を行う．3.1 項で作成した知識グラフにおいて，まず各実体 $e^i (0 \le i < |\mathbb{E}|; i \in \mathbb{N})$ に対し，5 個の名称 $e^i_j (0 \le j < 5; j \in \mathbb{N})$ を与える．$\mathbb{R}$ の元に対しても同様に 5 個の名称を与え，これらの名称を LM が扱う語彙とする．以降，同じ実体 (関係) に対する名称を「言い換え」と表現する．一文を 3 単語 (例: “$e^0_0\ r^1_1\ e^1_4$ ”) で構成したコーパスを作成し，初期化した 6，12，24 層の GPT-2 [16] と同様のアーキテクチャを持つモデルを学習させる．

推論時には 2 単語を入力し，出力された 1 単語が正しい実体を示す言い換えであるときに正解とする．学習の際，知識全体から言い換えの 20% をサンプリングしたものを訓練データとし，全ての言い換えへの汎化を期待する（詳細は付録 A を参照）．また，知識全体を全データとして，知識グラフを学習できているか評価する際に使用する．学習の結果，モデルは訓練データだけでなく，全データにおいても 99% 程の正解率を達成し，用意した知識を記憶できていることがわかる（表 1）．また，学習後の実体と関係の埋め込み表現に対して主成分分析を行う

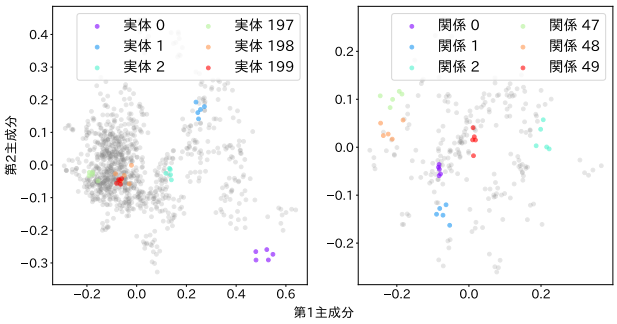

図 3　ER グラフを学習後の LM の埋め込み表現の主成分分析結果．左側は実体，右側は関係の埋め込み表現．各実体・関係はそれぞれ 5 つの言い換えをもち，同じ実体・関係に関する言い換えを同じ色で図示する（ここではそれぞれ 6 つの実体・関係に着目する）．主成分分析の結果より，言い換えの埋め込み表現は集まって分布しており，LM は言い換えを認識していると考えられる．

と，モデルが「言い換え」を認識していることが示唆される（図 3）．

3.3　編集手法：ROME

ROME [9] とは Causal LM に対する主要な知識編集手法の一つである．ROME では以下の 2 つのステップによってモデルの重み行列を更新し，特定の知識の編集を実現する．

Step 1: Causal Tracing　モデルが知識を連想するプロセスにおいて重要な役割を果たすモデル構成要素を特定する．これは，知識に関する情報を予測する[3]際のモデルの各隠れ状態の寄与を分析することで実現する（詳細は付録 B を参照）．Causal Tracing の結果，初期のフィードフォワード（FF）層が知識の連想に大きく寄与していることが示され，これは FF 層がキー・バリューメモリとして知識を格納していることを示した研究 [17] に裏付けられる．

Step 2: Rank-One Model Editing　Causal Tracing で特定した FF 層（式 1）における 2 層目の重み W_2 に対してランクが 1 の行列を足し合わせる．

$$\mathrm{FFN}(x) = \sigma(xW_1 + b_1)W_2 + b_2 \tag{1}$$

具体的には，W_2 を既存のキー・バリューのペア (K, V) に対する連想メモリとしてみなし，新しいキー・バリューのペア (k_*, v_*) を挿入するような編集を行う[4]．これは制約付き最小二乗問題に帰着し，更新後の重み $\hat{W}_2$ は次のようになる．

$$\hat{W}_2 = W_2 + \Lambda(C^{-1}k_*)^\top \tag{2}$$

ここで，$C^{-1} = KK^\top$ は K の非中心化された共分散であり，Λ は新しいキー・バリューペア (k_*, v_*) の残差エラーに比例するベクトルである．

3.4　知識の削除

LM における一般的な知識編集とは，(s, r, o) という知識を (s, r, o^*) のように，実体を新たな別の実体に関連付けることであると定義される．編集の一種で，(s, r, o) という知識を $(s, r, e_{\text{deleted}})$ のように，削除されたことを表す実体を用意してそれに関連付けるよう編集することを特に知識の削除と定義する．本研究ではこの知識の削除に特に焦点を当てる．

4　実験

手順　ある特定の知識を削除した際に，他の知識に対してどれだけ影響を与えているかを分析する．影響度は次ようにして求める．

1. 事前学習終了時の知識全体の正解率 $\mathrm{acc}_{\text{pre-del}}$ を計測する．
2. ある実体 e に関する知識を一つ削除する．つまり，(e, r, o_e) という知識を $(e, r, e_{\text{deleted}})$ となるような編集を行うことで知識を削除する．ここで，関係 r はある一つの関係に固定する．
3. 知識削除後のモデルに対して，知識全体の正解率 $\mathrm{acc}_{\text{post-del}}(e)$ を計測する．
4. ある実体 e に関する知識の削除前後の正解率の差を影響度 I として次のように求める．

$$I = \mathrm{acc}_{\text{pre-del}} - \mathrm{acc}_{\text{post-del}}(e) \tag{3}$$

結果・考察　図 4 に，各知識グラフを学習した 6 層モデルにおける，実体の次数と削除による影響度の関係を示す．図 4 より，ER グラフの場合については，次数の大きさと影響度の大きさに関係は見られないのに対し，BA グラフの場合については，次数の大きさと影響度の大きさに関係が見られることがわかる．すなわち，実世界とは異なる構造の ER グラフにおいては，各実体に関する知識を削除することによる影響度に差は見られず，実世界に似た構造の BA グラフにおいては，次数の高い実体に関する知識の削除はより大きな影響度を持ち，次数の低い実体に関する知識の削除はあまり影響度を持たないことがわかる．尚，影響度の絶対値は小さいが，実際の LM が格納する知識は膨大な量であるため，

3）　ここでは，$x = (s, r)$ という入力に対して o を予測することを指す．
4）　$K = [k_1|k_2|\ldots]$ および $V = [v_1|v_2|\ldots]$ であり，k と v はベクトルを表す．

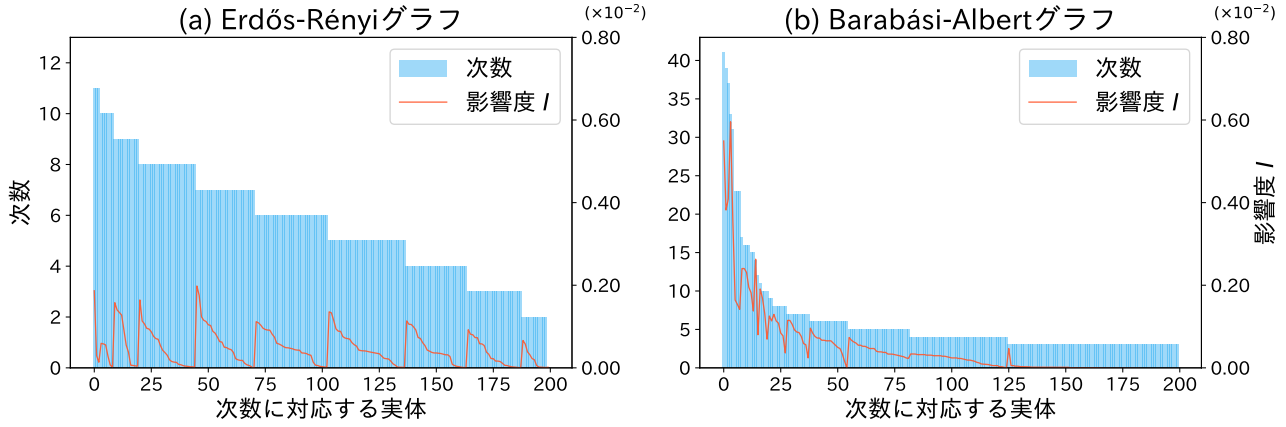

図 4　各知識グラフを学習した 6 層 GPT モデルにおける，実体の次数と削除による影響度の関係．左の縦軸は実体の次数を示し，横軸はそれに対応する実体を表している．右の縦軸は，その次数を持つ実体を削除した際に他の知識に与える影響度（副作用の度合い）を示す．ある実体に関する知識削除において，(a) ER グラフについては，次数と影響度の関係は見られないの対し，(b) BA グラフについては，次数と影響度の関係があることがわかる．影響度は他の知識への副作用の度合いを示すため，実世界に近い知識構造を学習させた LM において，頻出実体に関する知識削除の副作用は破滅的であることが示唆される．

表 2　各知識グラフを学習した各モデルにおける実体の次数と影響度の相関係数．

知識グラフ	モデル	相関係数
ER	6 層	.2093
	12 層	.2634
	24 層	.3045
BA	6 層	.9348
	12 層	.9892
	24 層	.7540

表 3　各知識グラフを学習した各モデルにおける実体の影響度 $I[10^{-2}]$ の統計値．

知識グラフ	モデル	最大値	最小値	平均値	標準偏差
ER	6 層	0.20	0.00	0.05	0.04
	12 層	0.20	0.00	0.07	0.04
	24 層	0.21	0.00	0.07	0.05
BA	6 層	0.60	0.00	0.05	0.08
	12 層	1.08	0.00	0.09	0.18
	24 層	1.13	0.00	0.09	0.17

知識全体への影響は大きいといえる．影響度はその実体に関する知識を削除することで他の知識にどれだけ副作用が生じているかを示している．そのため，仮説の通り，**重要な知識（次数の大きい実体に関する知識）の削除による副作用は大きく，重要でない知識（次数の小さい実体に関する知識）の削除による副作用は小さい**ことが示唆される．

また，他のモデルについての分析として，表 2 に各知識グラフを学習した各モデルにおける実体の次数と影響度の相関係数，表 3 に各知識グラフを学習した各モデルにおける実体の影響度の統計値を示す．表 2，表 3 より，12 層および 24 層のモデルについても，6 層のモデルと同様に，ER グラフにおける次数と影響度の相関係数の値が小さく，BA グラフにおける次数と影響度の相関係数の値が大きいことがわかる．これより，LM は学習させる知識グラフの構造の違いによって，知識削除における振る舞いが異なることが示され，**知識構造に着目して分析を行う重要性**が示唆される．

5　おわりに

本稿では，人工知識グラフを知識として持つ LM に対する知識の削除に関する分析を行い，頻出実体に関する知識削除の副作用が破滅的であることを明らかにした．また，人工的な知識グラフにおいても，実世界の知識と同様に，LM は実体どうしの関係や実体・関係の言い換えなどを認識していることが示され，人工データを用いて LM の知識をコントロールして分析を行う研究の有効性が示唆される．

今後は，異なる知識グラフの構造や知識編集手法においても同様の分析を進めていく．また，本研究では ROME を用いた知識削除を行ったが，削除した知識に関する痕跡が残っていないかについての詳細な分析も今後行っていく予定である．さらに，知識の完全な削除における課題に焦点を当て，人工的な知識グラフを用いて LM の知識をコントロールし，知識が削除可能な条件などについて分析することも興味深い．

謝辞

本研究は JSPS 科研費 JP22H00524，JP21K21343，JP21K17814 の助成を受けたものです．

参考文献

[1] Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. Language models as knowledge bases? In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, 2019.

[2] Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. How can we know what language models know? **Transactions of the Association for Computational Linguistics**, 2020.

[3] Benjamin Heinzerling and Kentaro Inui. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, 2021.

[4] Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A Smith, Yejin Choi, and Kentaro Inui. Real-Time QA: What's the answer right now? **arXiv preprint arXiv:arXiv:2207.13332**, 2022.

[5] Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. Are large pre-trained language models leaking your personal information? In **Findings of the Association for Computational Linguistics: EMNLP 2022**, 2022.

[6] Zhangyin Feng, Weitao Ma, Weijiang Yu, Lei Huang, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. Trends in integration of knowledge and large language models: A survey and taxonomy of methods, benchmarks, and applications. **arXiv preprint arXiv:2311.05876**, 2023.

[7] Ningyu Zhang, Yunzhi Yao, Bozhong 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 preprint arXiv:2401.01286**, 2024.

[8] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, 2022.

[9] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2022.

[10] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass editing memory in a transformer. **The Eleventh International Conference on Learning Representations (ICLR)**, 2023.

[11] Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. **arXiv preprint arXiv:2308.08742**, 2023.

[12] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, 2023.

[13] Yoichi Ishibashi and Hidetoshi Shimodaira. Knowledge sanitization of large language models. **arXiv preprint arXiv:2309.11852**, 2023.

[14] P. Erdös and A. Rényi. On random graphs i. **Publicationes Mathematicae Debrecen**.

[15] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. **science**, 1999.

[16] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[17] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, 2021.

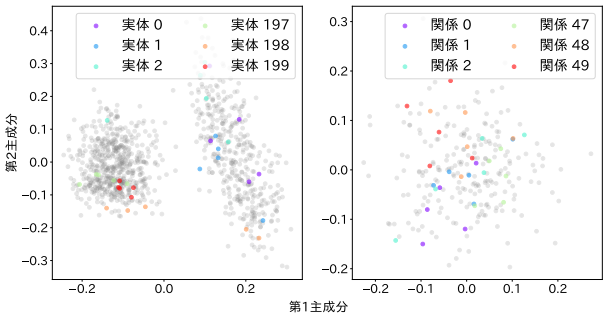

図 5　サンプル数を 2% に減少させ，ER グラフを学習させた LM の埋め込み表現の主成分分析結果．

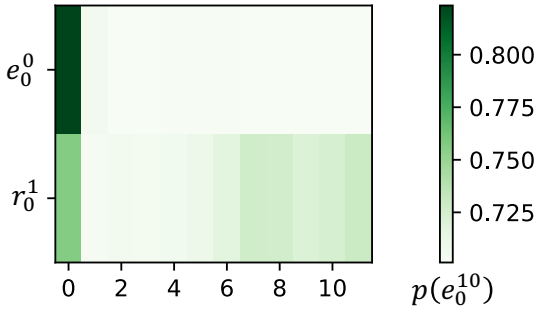

図 6　Causal Tracing による各層の FF 層の寄与度の分析結果の例（12 層モデルの場合）．横軸は LM の層，縦軸は各入力トークンに対応する．値は corrupted-with-restoration run 前後での正解トークンの生成確率の差を表す．色の濃い部分に対応する FF 層を修復することで，正解トークンを再び生成できるようになっており，その FF 層は知識の予測に寄与していることがわかる．

A　LM における言い換え表現の認識

3.2 項で示した通り，本研究で作成した人工知識グラフは，より実世界の設定に近づけるために，各実体・関係が言い換え表現を持つような構造を持つ．この知識グラフを学習させる際に知識をサンプリングして訓練データとして用いたが，これは全ての関係知識を LM に学習させてしまうと，それぞれの関係知識をただ暗記してしまい，言い換え表現を認識できないと考えられるからである．

3.2 項の図 3 の埋め込み表現は，知識グラフ全体の 20% をサンプリングして学習させた場合における結果であり，各実体・関係の言い換え表現の埋め込み表現が集まって分布している．一方で，サンプル数を減少させて学習させた場合においては，各実体・関係の言い換え表現の埋め込み表現が分散して位置していることがわかる（図 5）．これより，LM は初めから言い換え表現を認識しているのではなく，適切なサンプル数でサンプリングした訓練データを用いることで言い換え表現を認識する能力を獲得できることが示唆される．

B　知識編集手法の補足

3.3 項では，本研究の実験で使用する既存の知識編集手法 ROME について簡単に説明した．本節では，LM が知識を連想する際に重要な役割を果たす部分を特定するステップである Causal Tracing についてもう少し詳細に説明する．Causal Tracing は次のような手順によって，推論時における LM の各隠れ状態の寄与度を分析する[5]．

1. **clean run**：まず，知識に関する予測を行い，通常のモデルの隠れ状態を求める．具体的には，入力 $x = (s, r)$ からモデルが o を予測する際の全ての隠れ状態 $\{h_i^{(l)} \mid i \in [1, T], l \in [1, L]\}$ を求める．ここで，T は入力 x の長さ（この場合 $T = 2$），L はモデルの層数である．
2. **corrupted run**：次に，知識に関する予測を行う際に，subject に関する情報を隠すことで，破損したモデルの隠れ状態を求める．具体的には，x を入力した際に，subject に対応する埋め込み表現 $h_1^{(0)}$ に対してノイズを付加する（$h_1^{(0)} := h_1^{(0)} + \epsilon$）．その後，知識の予測を行い，破損した隠れ状態 $\{h_{i*}^{(l)} \mid i \in [1, T], l \in [1, L]\}$ を求める．これによって，clean run で出力できていた正しい知識を，corrupted run では出力できなくなる．
3. **corrupted-with-restoration run**：最後に，corrupted run で得られた破損した隠れ状態を持つモデルに対し，特定の隠れ状態 $h_{i*}^{(l)}$ を clean run で得られた通常の隠れ状態 $h_i^{(l)}$ に修復する．これを全ての隠れ状態に対してそれぞれ行い，知識の予測を行う．ある特定の隠れ状態を修復することで正しい知識を再び出力できるようになったとき，その隠れ状態は知識の予測に寄与していることがわかる．

図 6 に Causal Tracing による，各層の FF 層の寄与度の分析の結果例を示す．この場合，LM が入力トークン列 (e_0^0, r_0^1) から正解のトークン e_0^{10} を出力する際に重要な部分は，最初の層の FF 層であることがわかる．このようにして，Causal Tracing は，LM の知識予測の際に重要な役割を果たす部分の特定を実現する．

5）　LM の各隠れ状態に対する操作を，FF 層やアテンション層の場合についても考えることができる．