LLM 内部表現における予測可能成分と新規成分の分解

関口正登 [1]　石垣龍馬 [1]　前田英作 [1]
[1] 東京電機大学
{25amj16@ms, 24amj02@ms, maeda.e@mail}dendai.ac.jp

概要

LLM の内部表現には「過去のトークンから決まる文脈情報」と「その位置で入力されたトークンによる新規情報」とが混在する．この混在により，線形プローブなどによる分析で観測される差異が，文脈由来なのか入力トークン由来なのかを切り分けることが難しい．本研究では，重みを固定した LLM の内部表現（$h_{\leq t}$）から，次トークン位置の内部表現（h_{t+1}）を予測する外部予測器を学習し，その予測（$\hat{h}_{t+1}$）を予測可能成分，残差（$r_{t+1} = h_{t+1} - \hat{h}_{t+1}$）を新規成分として分解する枠組みを提案する．実験では，層およびトークン位置ごとに予測可能性を系統的に評価した．さらに線形プローブにより，（$\hat{h}_{t+1}$）は文書ドメイン分類に，（r_{t+1}）は次トークン種類分類に有利であることを確認し，両者には異なる情報が線形に埋め込まれている可能性を示唆した．

1　はじめに

大規模言語モデル（LLM）の内部表現を理解することは，モデルの挙動検証，失敗要因の特定，制御可能性や信頼性の向上において重要である [1]．しかし，自己回帰言語モデルにおける各トークン位置の内部表現には，それまでの文脈を要約した状態と，そのトークン位置で新たに入力されたトークンによる更新が同時に混在する．その結果，LLM の内部表現に含まれる情報の解釈を曖昧にし，線形プローブなどの内部表現の分析で観測される差異が文脈に影響されているのか，その位置のトークンに影響されているのかを判別することが困難である．そのため，ある層・ある位置の内部表現が，過去のトークンに依存した文脈情報と，そのトークン固有の情報のどちらに由来しているのか分解する必要がある．

本研究では，重みを固定した LLM の内部表現 $h_{\leq t}$ から，次トークン位置の内部表現 h_{t+1} を予測する外部予測器を学習し，その予測 $\hat{h}_{t+1}$ を「予測可能成分」，残差 $r_{t+1} = h_{t+1} - \hat{h}_{t+1}$ を「新規成分」と定義する．この枠組みにより，内部表現に混在する「過去のトークンから決まる文脈情報」と「その位置で入力されたトークンによる新規情報」を分解し，それぞれが保持する情報を切り分けて分析する．

2　関連研究

内部表現の予測可能成分と新規成分の分解

内部表現の分析において，トークン位置方向の構造を明示的に扱う重要性が指摘されている．Temporal Feature Analysis[2] は，LLM の内部表現がトークン位置方向の依存構造を持っている一方，スパースオートエンコーダ（SAE）[3, 4] が暗黙に時間方向の独立性や定常性を仮定している点を問題視し，内部表現を「文脈から予測可能な成分」と「文脈では説明できない新規成分」に分解する手法を提案した．本研究はこの観点に沿って，外部予測器によって内部表現を予測可能成分と新規成分に分解し，線形プローブを用いてそれぞれの成分がどのような情報を含んでいるか明示的に分析する．

内部表現の予測不可能性による複雑性の測定

PHi[5] は，LLM の出力に対する次トークン損失では，ランダム文字列のような「予測不能のため損失が高い入力」と，数学の問題のような「難易度が高いため損失が高い入力」を区別しにくい点を指摘し，次トークンの内部表現の予測可能性を複雑性の指標として測る枠組みを提案した．本研究では，内部表現の予測可能性という観点に着想を得て，外部予測器の残差 r_{t+1} が，次トークン予測の難易度とどの程度対応するかを分析する．

3　提案手法

重みを固定した LLM にトークン列 $x_{1:T}$ を入力したとき，ある層 ℓ の内部表現を $h_t^{(\ell)} \in \mathbb{R}^{d_{model}}$　$(t = 1, \ldots, T)$ とする．以後，層 ℓ を固定して $h_t := h_t^{(\ell)}$ と書く．

3.1 外部予測器

外部予測器 f_θ として，1 層の因果マスク付き multi-head self-attention を学習する．query/key/value に過去の内部表現 $\{h_1, \ldots, h_t\}$ を用いて，

$$\hat{h}_{t+1} = f_\theta(h_{\le t}) \tag{1}$$

として次トークン位置表現の予測 $\hat{h}_{t+1}$ を得る．残差を $r_{t+1} = h_{t+1} - \hat{h}_{t+1}$ とする．

3.2 学習目的と評価指標

外部予測器は教師モデルを固定したまま学習し，目的関数は内部表現 h_{t+1} の次元 d_{model} で正規化した平均二乗誤差（MSE）とする．

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{\|h_{t+1} - \hat{h}_{t+1}\|^2}{d_{model}}\right] \tag{2}$$

また，MSE は内部表現のスケールに影響される．そこで，スケールに依存しない指標として決定係数 R^2 を用いる．評価データ上でターゲット表現の分散を Var とし，

$$R^2 = 1 - \frac{\mathrm{MSE}}{\mathrm{Var}} \tag{3}$$

と定義する．

3.3 $\|r_{t+1}\|$ と surprisal の相関

LLM の次トークン分布 $p(x_{t+1} \mid x_{\le t})$ に対して，次トークンの surprisal を

$$s_{t+1} = -\log p(x_{t+1} \mid x_{\le t}) \tag{4}$$

とする．この指標は，LLM が実際に次に現れるトークンに対してどの程度の尤度を割り当てたかを負の対数尤度に変換したものであり，LLM が次のトークンにどの程度「驚いた」かを定量化する指標である．残差 r_{t+1} が内部表現の予測の難しさと結びついているか検証するため，$\|r_{t+1}\|$ と surprisal s_{t+1} の Spearman の順位相関係数を算出する．

3.4 線形プローブ：h，$\hat{h}$，r，$[\hat{h}, r]$ の比較

各位置の表現からタスクラベルを線形分類器で予測する線形プローブを学習する．入力表現は，h_{t+1}，$\hat{h}_{t+1}$，r_{t+1}，$[\hat{h}_{t+1}, r_{t+1}]$（concatenate）の 4 種類とする．本研究では，以下の 2 タスクを検証する．

- **pile_subset_22**：入力されたテキストが，The Pile データセットの 22 種類のサブセットのうち，どのサブセットからサンプルされたものかを，22 クラス分類するタスク．
- **next_token_type_8**：ある位置のトークンが，改行/空白/英字/数字/句読点・記号/開き括弧/閉じ括弧/その他のうち，どれに分類されるのかを，8 クラス分類するタスク．

4 実験設定

4.1 分析対象のモデル・層

分析対象の LLM として，EleutherAI/pythia-410m-deduped[6] を用いる．全 24 層（$\ell \in \{0, \ldots, 23\}$）のうち，$\ell \in \{0, 6, 12, 18, 23\}$ を対象とし，各層ごとに外部予測器を独立に学習した．

4.2 外部予測器の学習設定

外部予測器として，head 数が 32，query/key/value が 128 次元の attention を採用した．この設定は，Pythia-410m 内部の attention の 1 層を 4 倍した容量である．また，学習データには The Pile deduplicated[7] を用いて，入力として長さ 256 のトークン列に整形した．外部予測器は AdamW，バッチサイズ 32 で最適化し，Warmup と Cosine decay を用いて 30,000 ステップ学習を行い，100 ステップごとに評価した．

4.3 トークン位置ごとの分析

トークン位置ごとの分析では，学習済みの外部予測器に評価用データ 8,192 件を入力して各位置ごとの MSE および $\|r\|$ と surprisal の Spearman 相関係数を評価した．

4.4 線形プローブ

The Pile deduplicated から長さ $T = 256$ のトークン列（window）をサンプリングし，それぞれのタスクに対応するデータセットを作成した．`pile_subset_22` は 22 クラスのクラスバランスを揃えるため，1 クラスあたり{train: 4000，val: 500，test: 500}件を上限として収集し，`next_token_type_8` は，合計で{train: 40000，val: 5000，test: 5000}件収集した．また，よりシンプルな条件で比較するため，線形プローブの分析対象を最後のトークン位置に固定した．線形分類器として多クラスロジスティック回帰を採用し，タスク不均衡によるバイアスを避けるため，重み付き交差エントロピーで最適化する．

5　結果・考察

5.1　外部予測器の学習

図 1 に学習ステップにおける層ごとの MSE を示す．$\ell = 6, 12, 18$ では学習初期に急速に改善し，その後は緩やかに収束する傾向が見られる．また，$\ell = 0, 23$ では，内部表現のスケールが他の層より小さく，MSE が相対的に低い値になっている．

図 2 に層ごとの R^2 を示す．学習終盤において，$\ell = 0$ では R^2 が低く，$\ell = 6, 12, 18$ では R^2 が 0.3 程度で，最終層である $\ell = 23$ では 0.4 程度である．

これらの学習曲線から，過去の内部表現から次の内部表現がある程度予測可能で，層によって予測可能性が異なることが分かる．また，R^2 の層依存は内部表現が保持する情報の違いを反映していると考えられる．浅い層では次トークンに依存して表現が大きく変化しやすく，過去表現のみから一意に定まらない成分が多いため R^2 が低い可能性がある．一方，より深い層では文脈の要約や状態情報がより強く表現され，過去から再構成できる割合が増えることで R^2 が高くなると考えられる．

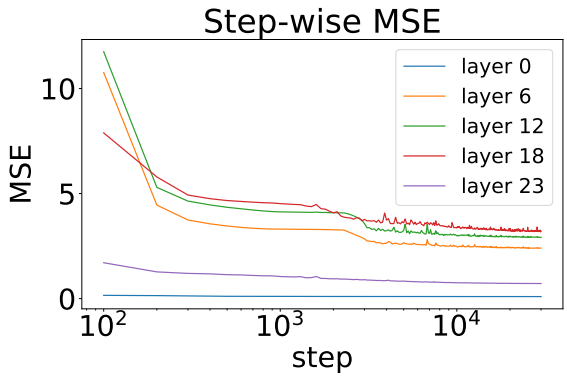

図 1: 外部予測器の MSE 学習曲線

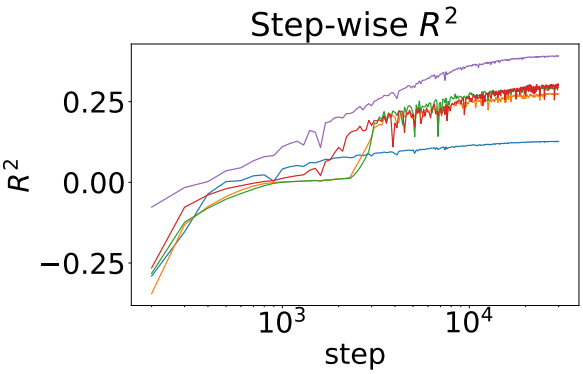

図 2: 外部予測器の R^2 学習曲線

5.2　トークン位置ごとの分析

図 3 に位置ごとの MSE を示す．最初の 32 トークンは MSE が極端に大きくなるため省略した．$\ell = 6, 12, 18$ ではトークン位置が後半になるにつれ，MSE が下がる傾向が見られる．一方，$\ell = 0, 23$ では，トークン位置に依存せずほぼ一定の値をとっている．

この結果から，中層（$\ell = 6, 12, 18$）では文脈が蓄積するほど次トークン位置表現の予測が容易になり，予測可能な成分が増えることが示唆される．一方，$\ell = 0$ および $\ell = 23$ では位置による変化が小さく，表層に近い表現（$\ell = 0$）や出力付近の表現（$\ell = 23$）が，文脈長の増加に対して相対的に安定している可能性がある．

図 4 に位置ごとの相関（Spearman：$\|r\|$ と surprisal）を示す．$\ell = 0$ および $\ell = 23$ では相関係数が相対的に高く，残差 $\|r\|$ のノルムが surprisal と相関していることが分かる．一方，$\ell = 6, 12, 18$ では相関が相対的に低く，特に $\ell = 18$ では後半で 0 付近まで低下する．また，位置依存の形状にも差がある．$\ell = 0$ では位置が進むにつれて相関が低下するのに対し，$\ell = 23$ では文脈が蓄積するにつれて相関が上昇する傾向が見られる．

5.3　線形プローブ

pile_subset_22 の結果を図 5，next_token_type_8 の結果を図 6 に，層（$\ell = \{0, 6, 12, 18, 23\}$）× 入力表現（$\{h, \hat{h}, r, [\hat{h}, r]\}$）の macro-F1 をヒートマップで示す．

pile_subset_22

pile_subset_22 では，残差 r よりも予測 $\hat{h}$ の方が高い性能を示した．すなわち，文書ドメインの識別に有用な情報は，次トークンの観測に強く依存する更新成分よりも，過去文脈から予測できる成分に強く現れることが分かる．

next_token_type_8

next_token_type_8 では，予測 $\hat{h}$ よりも残差 r の方が高い性能を示した．次トークンの表層的な特徴は，過去文脈のみからは一意に定まらない場合が多い．そのため，次のトークン位置への入力により確定した差分が r に強く反映され，線形に読み出しやすいと考えられる．

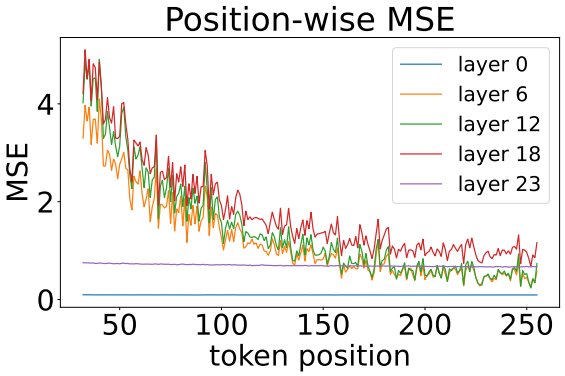

図 3: トークン位置ごとに算出した MSE

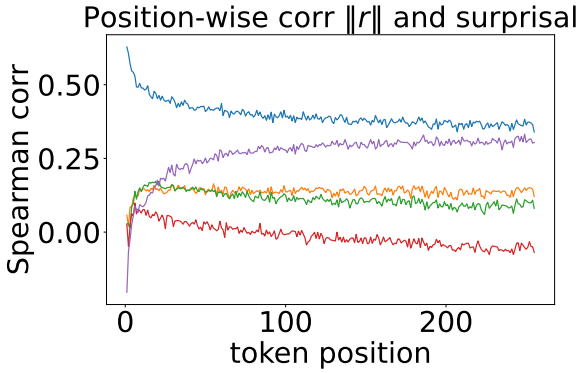

図 4: $\|r_{t+1}\|$ と surprisal の Spearman 相関

$\hat{h}$ からは文脈に依存する特徴を読み出しやすく，r からは次トークンそのものの特徴を読み出しやすいことから，$\hat{h}$ と r には異なる種類の情報が線形に埋め込まれていると考えられる．この結果は，内部表現を「予測可能成分」と「新規成分」に分解できる可能性を示唆する．

6　結論・今後の課題

6.1　結論

本研究では，重みを固定した LLM の内部表現に対し，外部予測器として 1 層の因果マスク付き multi-head self-attention を学習し，次トークン位置表現を予測可能成分 $\hat{h}$ と残差 r に分解して分析した．層スイープ（0，6，12，18，23）により，次トークン位置表現は過去表現から一定程度予測可能であり，予測可能性（R^2）が層によって異なることを示した．また，残差 r のノルムと surprisal の Spearman 相関は層・位置に依存して変化し，特に $\ell = 0$ と $\ell = 23$ で相対的に高い相関が観察された．さらに線形プローブでは，$\hat{h}$ が文脈に依存する情報および r が次トークン固有の情報が線形に埋め込まれている可能性を示唆した．

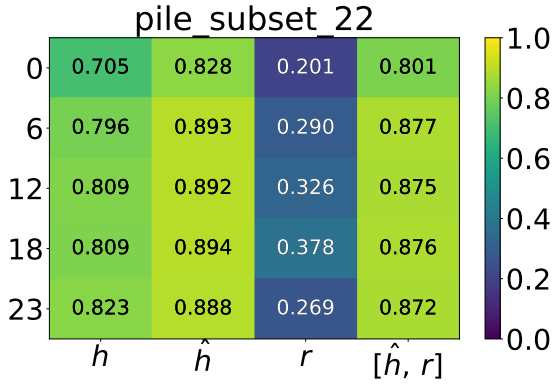

図 5: pile_subset_22 の micro-F1 スコア
層 ×入力表現のヒートマップ

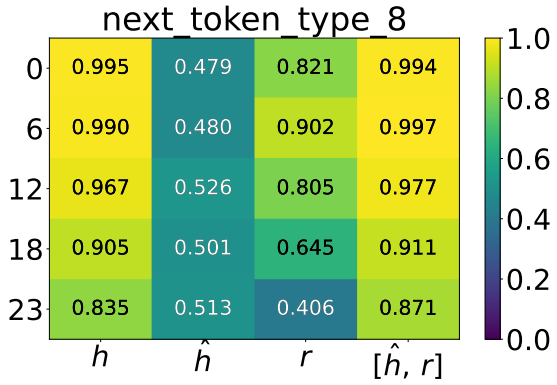

図 6: next_token_type_8 の micro-F1 スコア
層 ×入力表現のヒートマップ

6.2　今後の課題

今後の課題として，外部予測器のアーキテクチャ依存性の検証が挙げられる．本研究では外部予測器を 1 層 attention に固定したが，層数・ヘッド数・非線形性などの扱いを変えた場合に $\hat{h}$ と r の分解がどの程度変化するかを調べる必要がある．また，LLM のモデルサイズやアーキテクチャを変えた際の再現性を検証したい．そして，本研究における内部表現の分解によって得られた表現が因果的に有効であることを示すため，$\hat{h}$ および r を用いた activation steering によって LLM の生成がどのように変化するかを検証したい．

参考文献

[1] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety – a review. **arXiv preprint arXiv:2404.14082**, 2024.

[2] Ekdeep Singh Lubana, Can Rager, Sai Sumedh R. Hindupur, Valerie Costa, Greta Tuckute, Oam Patel, Sonia Krishna Murthy, Thomas Fel, Daniel Wurgaft, Eric J. Bigelow, Johnny Lin, Demba Ba, Martin Wattenberg, Fernanda Viegas, Melanie Weber, and Aaron Mueller. Priors in time: Missing inductive biases for language model interpretability, November 2025.

[3] Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. **arXiv preprint arXiv:2309.08600**, 2023.

[4] Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. **arXiv preprint arXiv:2406.04093**, 2024.

[5] Vincent Herrmann, Róbert Csordás, and Jürgen Schmidhuber. Measuring in-context computation complexity via hidden state prediction, March 2025.

[6] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, **Proceedings of the 40th International Conference on Machine Learning**, Vol. 202 of **Proceedings of Machine Learning Research**, pp. 2397–2430. PMLR, 23–29 Jul 2023.

[7] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, December 2020.