誤字に対する Transformer ベース LLM のニューロンおよびヘッドの役割調査

辻 航平 [1]　平岡 達也 [2]　鄭 育昌 [1,3]　荒牧 英治 [1]　岩倉 友哉 [1,3]
[1] 奈良先端科学技術大学院大学　[2]MBZUAI　[3] 富士通株式会社
tusji.kohei.tl1@naist.ac.jp
tatsuya.hiraoka@mbzuai.ac.ae
aramaki@is.naist.jp
{cheng.yuchang,iwakura.tomoya}@fujitsu.com

概要

本論文では，FFN 層のニューロンや，アテンション層のアテンションヘッドが誤字を認識・修復しているという仮説を立て，誤字を含む文が入力されたときに活発に働く，**誤字ニューロン**および**誤字ヘッド**を特定する．我々の実験結果から以下のことが判明した．1) 初期層と中間層前半に誤字の認識と修復を行うニューロンが存在し，中間層前半にあるニューロンが誤字の修復の中核である．2) 広く文脈情報を捉えるヘッドが誤字の修復に貢献している．3) 誤字ヘッドの中には，単語の意味的な繋がりを認識するヘッドが存在した．

1　はじめに

大規模言語モデル (LLM) は広く使われており [1]，入力に誤字が含まれている可能性もある．LLM が誤字を “修復” して，誤字を含んでいても正しい推論を行うことも多い [2] が，誤字によって LLM が “損害” を受け，間違った推論を行う場合もある [3]．

実際に図 1 が示すよう，誤字が多い場合には，大きなモデルほど性能を維持しているものの，性能低下が発生している．このような誤字による損害を軽減するためには，誤字に対する頑健性と誤字による性能劣化の原因をより深く理解する必要がある．

誤字に関する既存研究は，摂動に対する頑健性を測るデータセットの作成 [3] や，頑健性を向上させるための工夫 [4] が主である．誤字がどの層で修復されるかを調査した研究 [5] も存在するが，単語のみを入力とし，層単位の調査のみが行われている．

我々は，誤字に対する頑健性は Transformer [6] アーキテクチャの主要部分である Feed-Forward Network (FFN) 層およびアテンション層の内部挙動によってもたらされると仮定し，検証する．具体的には，FFN 層の 2 つの線形層の間の活性化関数の出力 “ニューロン” [7] と，アテンション層の各アテンションヘッドの働きを見ていく．これらには，特定のタスク [8] や知識 [9, 10]，挙動 [11, 12] を促進するものが報告されている．これらの中で特に誤字の認識・修復を行う**誤字ニューロン**や**誤字ヘッド**を特定することを目的とする．本研究では，文脈を用いることができる環境での誤字に対する内部挙動を調査するために，単語特定タスク (§2) を用いる．Gemma 2 [13] を用いた実験の結果から，以下のことが示唆される．

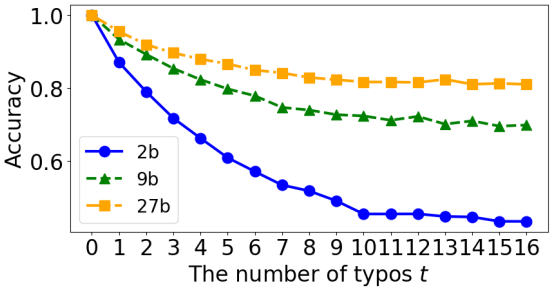

図 1: §2.2 において作成したデータセットに t 個の誤字を付与した場合の精度．

- 初期層と中間層前半に誤字の認識と修復を行うニューロンが存在し，中間層前半にあるニューロンが誤字の修復の中核である (§3)．
- 特定のトークンに注目するヘッドではなく，広く文脈情報を捉えるヘッドが誤字の修復に貢献している (§4.2)．
- 誤字ヘッドの中には，単語の意味的な繋がりを認識するヘッドが存在した (§4.3)．

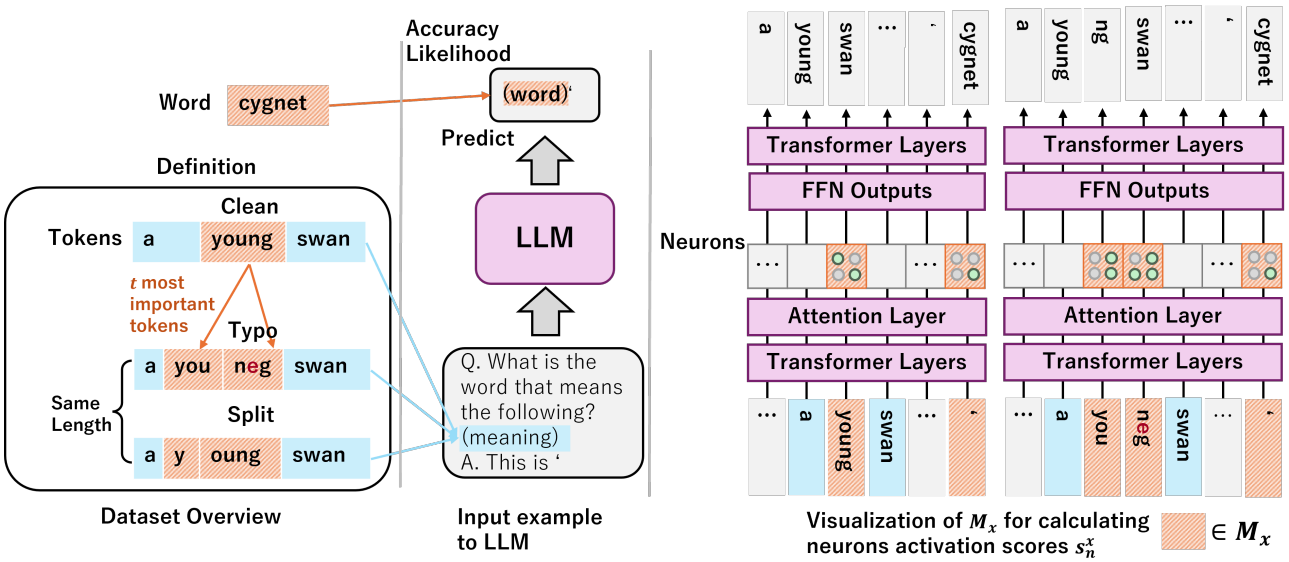

図 2: データセットの概要 (左)，LLM への入力例 (中)，ニューロンの活性化スコア s_n^x を計算するための M_x の可視化 (右).

2 準備

2.1 モデル

Gemma 2 [13] の 2B，9B，27B モデルを用いた．27B モデルのみ bfloat16 で読み込み，他は float32 で読み込んだ[1)]．また，生成には貪欲法を用いた．

2.2 誤字のないクリーンデータセット

文脈を考慮した誤字の影響を調べるために，与えられた語義に対応する単語を出力させる単語特定タスクを利用する．例えば，“*a young swan*” が語義として入力された場合には，対応する単語 “*cygnet*” を出力すれば正解となる．[14] に倣い，62,643 組の単語-語義ペアを WordNet [15] から抽出した[2)]．また，図 2 中央に示すように，プロンプトを設計した．

誤字がない場合に LLM が正答できるようにするため，モデルごとに正答できるデータ 5,000 組の単語ー語義ペアを抽出し，データセットとした．

2.3 誤字の付与

先行研究 [3] に倣って，Gemma 2 2B でタスクを解く際に重要なトークン t 個を逆伝播により決定し，それらにランダムな 1 文字を追加することで，誤字データセットを作成する (図 2 左).

誤字を含む入力のトークン系列は，誤字のない入力のトークン系列と異なるトークン数になる場合が多い．例えば，“young” は 1 トークンだが，誤字を含む “youneg” は “you / neg” と 2 トークンになる．これらの比較では，トークン数の違いによる影響を受ける．トークン数の違いによる内部挙動の差を除くため，誤字データと同じ長さのトークン化候補を選択したデータで構成される分割データセットも作成した (図 2 左).

3 誤字ニューロン

3.1 誤字ニューロンの特定手法

先行研究 [11] に倣い，各データセットで活性化しているニューロンの差を比較して，誤字にのみ反応するニューロンの存在を明らかにしていく．トークン系列 $x = w_1, ..., w_m, ..., w_{|x|}$ のトークン長を $|x|$ とすると，データセット $X \ni x$ におけるニューロン n の活性化スコア s_n^X は次のように定義される：

$$s_n^X = \frac{1}{|X|} \sum_{x \in X} \left(\frac{1}{|M_x|} \sum_{m \in M_x} f(x_1^m, n) \right), \qquad (1)$$

ここで，$|X|$ はデータ数，$f(x_1^m, n)$ は，LLM が $x_1^m = w_1, ..., w_m$ を入力されたときの w_m におけるニューロン n の出力，M_x はトークン位置を示すインデックス集合，$|M_x|$ は M_x の個数である．本実験では M_x を，答えとなる単語の直前と t 個の重要語のトークン系列を示すインデックス集合とする．図 2 右のオレンジの入力が M_x を構成するトークンである．

ニューロンの誤字への重要度 Δ_n は次の式になる:

$$\Delta_n = s_n^{X_{\text{typo}}} - \max \left(s_n^{X_{\text{clean}}}, s_n^{X_{\text{split}}} \right), \qquad (2)$$

1) Xeon Gold 6230R + NVIDIA A100 40GB*2 を用いた．
2) NLTK [16] ver.3.9.1 に実装されている WordNet を用いた

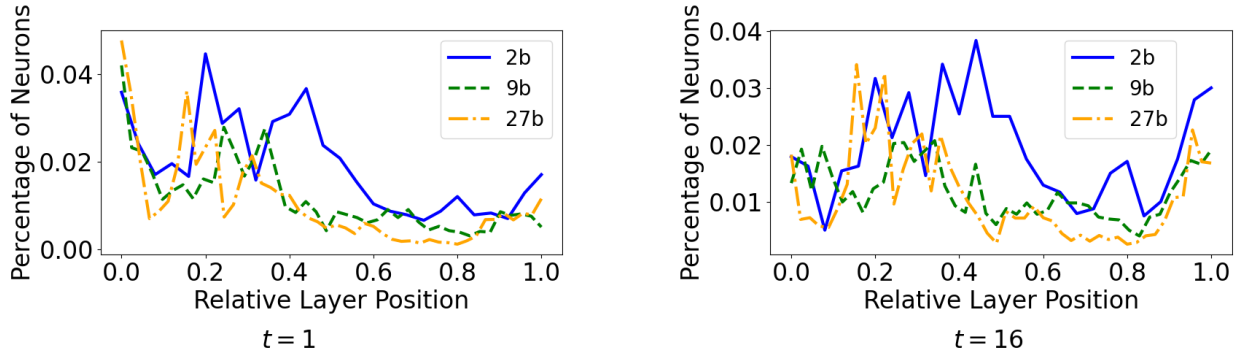

図 3: 層ごとの誤字ニューロンの割合．左が $t = 1$ の場合，右が $t = 16$ の場合．モデルサイズにより層の総数が異なるので，x 軸は 0 から 1 の相対位置で現した．

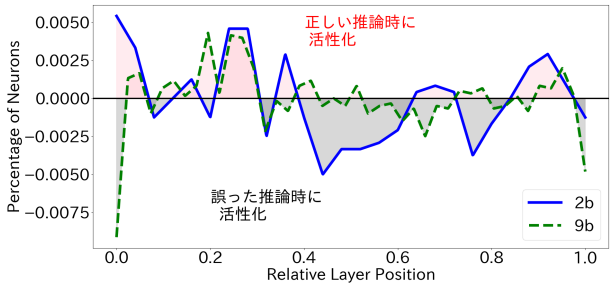

図 4: 誤字ニューロンの層ごとの分布．黒線より上の値は，LLM が正しい単語を予測したときに誤字ニューロンが多く活性化したことを示す．

X_{typo}，X_{clean}，X_{split} はそれぞれ誤字データセット，クリーンデータセット，分割データセットを示す．

Δ_n が大きいニューロン n は，誤字に特化しており，Δ_n 上位 K 個を誤字ニューロンとする．

3.2 結果

$t = 1, 16$ での誤字ニューロンの分布を図 3 に示す．Δ_n が上位 0.5%以内かつ $\Delta_n > 0$ のニューロンを誤字ニューロンとした．

誤字ニューロンは，$t = 1$ で初期層 (0.0〜0.2) に多く，$t = 1$ と $t = 16$ の両方で中間層前半 (0.2〜0.5) に多く存在する．中間層前半は文脈を考慮した処理を行っているとされている [17]．そのため，初期層で誤字を修復できなかった場合に，中間層前半が広い文脈を用いて修復していることが示唆される．

最終層付近の誤字ニューロンは，$t = 16$ の場合に多い．これは最終層付近の内部表現にも誤字による影響が残っているためだと考えられる．

3.3 誤字の修復に寄与するニューロン

§3.2 では，LLM が正しく推論しているかどうかを考慮していない．本節では，LLM が誤字を修復できた場合とできなかった場合での誤字ニューロンの活性化の違いに注目する．また，小型モデル (2B) と大型モデル (9B，27B) で，ニューロンの傾向が異なっていたため，2B と 9B のモデルを用いて実験した．

5,000 件のデータセットから，誤字による損害を受けず，正しい単語が予測された 100 件と，誤字による損害を受け，誤った単語予測につながった別の 100 件を抽出し，誤字ニューロンの活性化の違いを比較した．実験は $t = 1$ で行い，Δ_n 上位 0.5%の誤字ニューロンの層分布の差を調査した．

図 4 に結果を示す．9B モデルでは，損害を受けた場合に初期層の誤字ニューロンが増加する．これは，誤字の修復以外の役割の初期層のニューロンの誤った活性化が正しい認識を妨げている可能性がある．2B モデルでは，損害を受けた場合には中間層の誤字ニューロンが増加した．これは，2B モデルは，中間層への依存度が高いためだと考えられる．どちらのモデルにおいても，誤字を修復できた場合，中間層前半の誤字ニューロンが増加したため，中間層前半の誤字ニューロンは重要だとわかる．

4 アテンションヘッド

4.1 誤字ヘッドの特定手法

サブワード結合 [18] のように，誤字の修復がヘッドでも行われている可能性が高い．このようなヘッドは，誤字を含む入力でのみ，修復に重要なトークンに注目する，または文脈を見るために複数のトークンに注目が分散することが予想される．

そこで，アテンションマップの各行を確率分布とみなして一様分布との KL ダイバージェンスを計算することで誤字に特化したヘッドを特定する．KL ダイバージェンスはトークンの数に対して単調増加であり，誤字データセットや分割データセッ

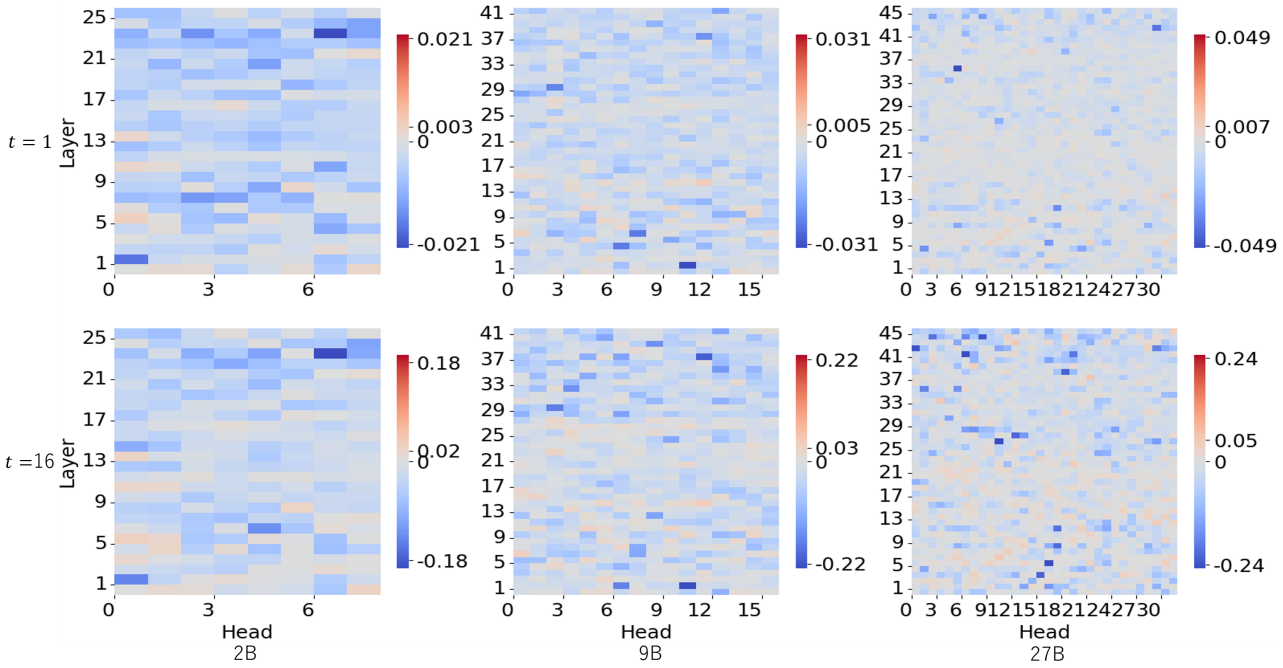

図 5: 各モデル，各誤字数ごとの Δ_h の分布

トはクリーンデータセットよりもトークンの数が多いため，数値が高くなりやすい．そこで，最大値 $\log_2 m$ で正規化した以下の式を用いた:

$$s_h^X = \frac{1}{|X|} \sum_{x \in X} \left(\sum_m \left(\frac{D_{\mathrm{KL}}(P_{x,m,h} || U_m)}{\log_2 m} \right) \right), \quad (3)$$

このとき，$D_{\mathrm{KL}}(\cdot)$ は KL ダイバージェンスを返す関数，U_m は m 個の確率変数を持つ一様分布，$P_{x,m,h}$ は入力 x に対するヘッド h のアテンションマップの m 行目である．

ヘッドの誤字への重要度 Δ_h は次の式になる:

$$\Delta_h = s_h^{X_{\mathrm{typo}}} - \max \left(s_h^{X_{\mathrm{clean}}}, s_h^{X_{\mathrm{split}}} \right), \quad (4)$$

X_{typo}，X_{clean}，X_{split} はそれぞれ誤字データセット，クリーンデータセット，分割データセットを示す．Δ_h の絶対値が大きいヘッドは，誤字を含む入力に対する挙動が，誤字を含まない入力とは大きく異なる．そのため，**Δ_h の絶対値が大きい上位 J 個のヘッドを誤字ヘッドとした．**

4.2 結果

$t \in 1, 16$ での Δ_h は図 5 のようになった．すべての設定で，最大値と最小値の絶対値の差は約 10 倍である．この結果から，ヘッドにおける誤字の認識と修復は特定のトークンへの集中ではなく，幅広く文脈をみることで行われていることがわかる．

4.3 誤字ヘッドの可視化

9B モデルの Δ_h の絶対値上位 1.5%のヘッドを誤字ヘッドとして，各入力に対するアテンションマップを観察した．Appendix A にヘッドの可視化の例を載せる．

初期層の誤字ヘッドは，文の切れ目を認識するヘッドである．中間層前半の誤字ヘッドは意味的な繋がりに反応しているヘッドであり，類語から誤字を修復していると考えられる．最終層付近のヘッドも，わずかながら誤字がある場合に広く文脈を見ることで修復していると考えられる．また，ほとんどの誤字ヘッドは'<bos>' に強い注意を向けていた．

5 おわりに

本研究では，Transformer ベースの LLM のニューロンやヘッドの誤字に対する反応を調査した．実験結果より，初期層と中間層前半の一部のニューロンが誤字に反応し，特に中間層前半のニューロンが，誤字の修復において重要であると判明した．また，広く文脈を見るヘッドが誤字を修復している．以上より，誤字に関する分析では初期層や中間層前半に着目することが重要であると言える．本研究では 1 トークンに 1 文字の人工的な誤字挿入に限定したが，将来的にはより現実的な誤字を考慮してゆく．

謝辞

本研究は,「戦略的イノベーション創造プログラム（SIP)」「統合型ヘルスケアシステムの構築」JPJ012425 および JST CREST「リアルワールドテキスト処理の深化によるデータ駆動型探」（課題番号：JPMJCR22N1）の支援を受けたものである.

参考文献

[1] Sumit Kumar Dam, Choong Seon Hong, Yu Qiao, and Chaoning Zhang. A complete survey on llm-based ai chatbots. **arXiv preprint arXiv:2406.16937**, 2024.

[2] Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, et al. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. **Data Engineering**, p. 48, 2024.

[3] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. **Advances in Neural Information Processing Systems**, Vol. 36, , 2023.

[4] Hongyi Zheng and Abulhair Saparov. Noisy exemplars make large language models more robust: A domain-agnostic behavioral analysis. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, **Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing**, pp. 4560–4568, Singapore, December 2023. Association for Computational Linguistics.

[5] Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. From tokens to words: On the inner lexicon of llms. **arXiv preprint arXiv:2410.05864**, 2024.

[6] A Vaswani. Attention is all you need. **Advances in Neural Information Processing Systems**, 2017.

[7] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[8] Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 11132–11152, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[9] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 8493–8502, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[10] Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. Successor heads: Recurring, interpretable attention heads in the wild. In **The Twelfth International Conference on Learning Representations**, 2024.

[11] Tatsuya Hiraoka and Kentaro Inui. Repetition neurons: How do language models produce repetitions? **arXiv preprint arXiv:2410.13497**, 2024.

[12] Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. Copy suppression: Comprehensively understanding a motif in language model attention heads. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen, editors, **Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP**, pp. 337–363, Miami, Florida, US, November 2024. Association for Computational Linguistics.

[13] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. **arXiv preprint arXiv:2408.00118**, 2024.

[14] Candida Maria Greco, Lucio La Cava, and Andrea Tagarelli. Talking the talk does not entail walking the walk: On the limits of large language models in lexical entailment recognition. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Findings of the Association for Computational Linguistics: EMNLP 2024**, pp. 14991–15011, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[15] Christiane Fellbaum. Wordnet and wordnets. In Keith Brown, editor, **Encyclopedia of Language and Linguistics**, pp. 2–665. Elsevier, 2005.

[16] Steven Bird and Edward Loper. NLTK: The natural language toolkit. In **Proceedings of the ACL Interactive Poster and Demonstration Sessions**, pp. 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[17] Vedang Lad, Wes Gurnee, and Max Tegmark. The remarkable robustness of LLMs: Stages of inference? In **ICML 2024 Workshop on Mechanistic Interpretability**, 2024.

[18] Javier Ferrando and Elena Voita. Information flow routes: Automatically interpreting language models at scale. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, **Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing**, pp. 17432–17445, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

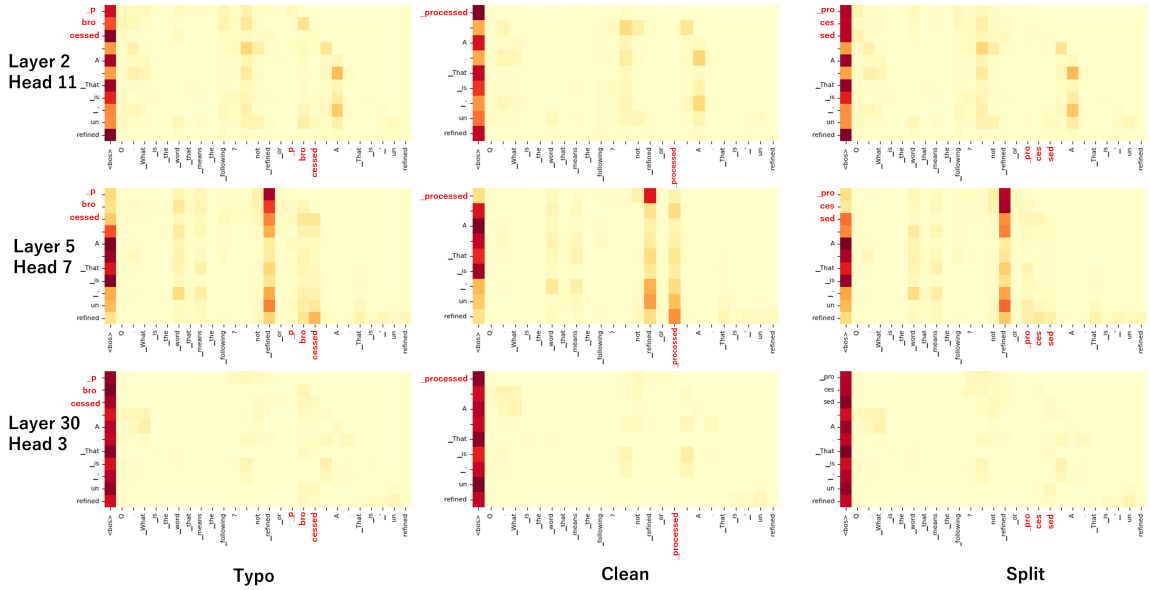

図 6: 2B モデルにおける誤字ヘッドの可視化．“_processed” および誤字を付与された “_pbrocessed” のトークン系列を赤色で示している．先頭のトークンは’<bos>’，Layer 5 Head 7 において’<bos>’ 以外に強い注意を向けらているトークンは’_refined’ である．

	2B		9B	
	Clean	Typo	Clean	Typo
Vanilla	1.00	0.86	1.00	0.93
⊖ Random Neurons	0.98	0.87	0.99	0.93
⊖ Typo Neurons	0.84	0.73	0.96	0.90

表 1: ニューロン切除 (⊖) を行った場合のクリーンデータセットと誤字データセットにおける精度．“Vanilla” はニューロン切除なしの精度．

	2B		9B	
	Clean	Typo	Clean	Typo
Vanilla	1.00	0.86	1.00	0.93
⊖ Random Heads	0.87	0.76	0.80	0.76
⊖ Typo Heads	0.91	0.80	0.89	0.84

表 2: ヘッド切除 (⊖) を行った場合の精度．

A　ヘッドの可視化の具体例

§4.3 の可視化の例を図 6 に示した．誤字のない入力は “<bos> / Q / . / _What / _is / _the / _word / _that / _means / _the / _following / ? / \n / not / _refined / _or / _processed / \n / A / . / _That / _is / _’ / un / refined” であり，誤字により “_processed” が “_pbrocessed” になる．

B　ニューロンの切除

データセット中のランダムな 100 件を用いて誤字ニューロンを特定し，それらを切除したうえで，残りの 4,900 件での精度を評価する．このとき，上位 0.5%のニューロンを誤字ニューロンとして，ベースラインとして 0.5% のランダムなニューロンを切除した場合でも精度を測った．ニューロンの出力値をゼロにすることで切除した．クリーンデータセットと $t = 1$ の誤字データセットに対して実験を行った．

表 1 に実験結果を示す．誤字データセットでは，ランダムな切除で性能は低下せず，誤字ニューロンの切除でのみ精度が低下したことから，少数の誤字ニューロンが誤字の修復を担っていることがわかる．クリーンデータセットに対しても，誤字ニューロンの切除がランダムな切除よりも性能が低下したため，誤字ニューロンは誤字に限らず，通常の処理においても役割を持っている可能性がある．

C　ヘッドの切除

Appendix B と同様の実験を誤字ヘッドに対しても行った．このとき，誤字ヘッドおよびランダムなヘッドの割合を 1.5%とし，該当ヘッドのアテンションスコアをすべて 0 にすることで切除した．

表 2 に実験結果を示す．どちらのモデルとデータセットにおいても，ランダムなヘッドの切除の方が，誤字ヘッドの切除よりも精度が低下している．これは，4.3 で述べたように誤字ヘッドの多くは’<bos>’ に集中し，通常の処理で他ヘッドと比較して役割が少ないためだと考えられる．また，ランダムなヘッドの切除でも誤字データセットの精度が，落ちていることから，誤字の修復はヘッド全体で薄く広く行われている可能性がある．