言語モデルが有する時間的推論に関する事実知識の分析

中屋和樹 [1] 松田源立 [1]
[1] 成蹊大学大学院 理工学研究科
dd246201@cc.seikei.ac.jp matsuda@st.seikei.ac.jp

概要

ニューラル言語モデルは, 大量のテキストを用いた事前学習と指示チューニングによって, 多様な知識と推論能力を獲得しているとされる. 本研究では, 言語モデルが有する時間に関する事実知識, 特に過去の出来事の日付に関する知識について調査する. 具体的には, 過去の出来事が発生した年月日を回答するタスクのデータセットを構築し, 言語モデルが出力する日付と実際の正解との間の誤差を分析する. 実験の結果, 言語モデルは年号に関する知識については, 月日と比較してより正確に保持していることがわかった. また, 年号を正確に認識できていない言語モデルは, 実際の年よりも後の年代として推論しやすい傾向があることが示唆された.

1 はじめに

ニューラル言語モデルの成功により, 自然言語処理の分野は大きな飛躍を遂げた. 現在の主流である Transformer[1] を基盤とした LLM は, 大量の Web テキストを用いた事前学習を行った後に, RLHF[2] などの強化学習によって人間の嗜好に合わせてアライメントされる. この過程でどのような知識や推論能力が獲得されるのかという問いは, 言語モデル研究における重要な課題となっている.

人間の基本的な認知能力の一つに時間的推論（Temporal Reasoning）がある. 時間的推論とは, 時間的な概念を理解し, 関係を推論する能力のことを指す. ChatGPT を含めた近年の LLM's は, この能力を一定程度有していると報告されているものの, その能力には限界が存在し, 依然として十分な時間的推論を行えていないという指摘も存在する [3][4].

LLM の時間的推論を測定するためのデータセットとして, TIMEBENCH[5] が提案されている. TIMEBENCH は, 3 つの時間的な概念から総合的に LLM の時間的推論を測ることを目的として設計されている. しかし, これは時間的な事実知識そのものの評価ではなく, 文脈に応じた時間的推論に主眼を置いたものとなっている. また, 日本語においても時間的推論を測ることを目的としたデータセットは存在しているが [6], 年代別の事象に焦点を当てて, それらの事実知識に関して言語モデルの体系的な調査を行った研究は確認できなかった.

上記の背景を踏まえ, 本研究では, 言語モデルの時間的な事実知識の保有能力に焦点を当てた調査を行う. 具体的には, 「〈イベント〉は何年の何月何日？」という質問を LLM に入力し, モデルが正確に西暦と月日を回答できるかどうかを定量的に分析する. 特に, 年代と正答率の関係を明らかにするため, 正答率を年代別に測定する. さらに, 分析対象をモデルの出力だけでなく内部モジュールにも拡張し, 一部モジュールを無効化した場合の出力変化についても調査を行う.

実験を通して, 以下の分析結果が得られた.

- 年号に関する知識の方が月日よりも定着している.
- 言語モデルはイベントの発生時期を実際よりも後の年代として予測しやすい傾向にある.
- 一部の MLP モジュールに集中して年月日の知識が蓄積されている可能性がある.

2 データセットの構築

本研究では, 「〜は何年の何月何日?」という質問に対して年号及び月日が回答となる, 事実 QA データセットの作成を行う. データセット作成にあたって必要となる情報は, 特定のイベントとそのイベントが発生した時間情報であるが, Wikipedia では「〜年の日本」というタイトルのページが作成されており, 本研究ではここから情報を抽出する. ページ数の都合上, データセットの詳細な作成方法については Appendix[A] で述べる. 主な手順としては, Wikipedia からの情報抽出→イベントの分割→ QA 形式への変形→人手によるスクリーニングとなっており, 表 1

に各フェーズで行った作業後に得られたデータ数を示した.

表 1 各フェーズでのデータ数の遷移

	抽出後	分割後	QA 作成後	選別後
データ数	5,815	6,774	5,606	5,296

3 事前実験

本研究で使用するデータセットの質問の語尾は「〜は何年の何月何日？」となっており, これに対応する言語モデルの出力は, 例えば, 1999 年 5 月 21 日のようなフォーマットに従っていることが望ましい. そこで, まずは指示チューニング言語モデルがこのような回答を出力する機能を有しているか調査するための事前実験を行った. 具体的には, 言語モデルに質問を入力し, 得られた出力に前述のフォーマットに従った部分文字列が 1 ヶ所存在した場合のみ有効回答とみなし, その割合を計算した. データセットには前節で収集したもの全てを使用し, 貪欲デコーディングで推論を行った. 該当箇所の抽出は, 正規表現によるマッチングを採用し, 1999 年（平成 11 年）5 月 21 日のように出力にブレが生じた場合でも対応可能にしている. また, 個々の出力について著者が目視での確認を行った結果, いくつかの言語モデルにおいて, 西暦を含んだ上で回答を拒否するケースが見られたため, このような場合には無効回答とみなした. 表 2 の数値は, 各言語モデルの出力に○年○月○日というフォーマットに従った部分文字列が含まれている割合（is date rate）を示したものである. 全てのモデルで is date rate の割合は 0.7 を超えており, 概ねフォーマット通りの回答が得られていることが示唆される. 次節の評価方法のセクションでは, この結果を踏まえた評価方法を検討する.

表 2 各モデルの is date rate

モデル名	is date rate
google/gemma-2-2b-jpn-it	0.87
rinna/gemma-2-baku-2b-it	0.71
llm-jp/llm-jp-3-1.8b-instruct	0.85
llm-jp/llm-jp-3-3.7b-instruct	0.73
llm-jp/llm-jp-3-13b-instruct	0.80
elyza/Llama-3-ELYZA-JP-8B	0.95
rinna/llama-3-youko-8b-instruct	0.94
cyberagent/calm3-22b-chat	0.80

4 実験設定

4.1 言語モデル

対話生成に特化した大規模言語モデルは, 大量のテキストによる事前学習を行ったのちに, 指示チューニング（SFT や PPO といった強化学習手法全般の総称とする）によって人間の嗜好に合わせたアライメントが施される. 本研究では, 年代別の事実知識の解答性能を調査する対象として, 指示チューニング後の言語モデルを選択する. 使用したモデルは次のとおり. google/gemma-2-2b-jpn-it[1], rinna/gemma-2-baku-2b-it[2], llm-jp/llm-jp-3-1.8b, 3.7b, 13b[3], elyza/Llama-3-ELYZA-JP-8B[4], rinna/llama-3-youko-8b-instruct[5], cyberagent/calm3-22b-chat[6], .

4.2 評価方法

言語モデルの出力の評価については, 様々な手法が存在するが, タスクに応じて適切な指標が用いられることが多い. 本研究では, 言語モデルの年代別の事実認識能力を評価するために, 出力に含まれる西暦, 月日を参照と比較する必要があるが, 出力テキストそのままでは exact match や char F1 を評価指標として適用することができない. また, llm-as-a-judge[7] のような別の大規模言語モデルを使用する手法は, 計算コストが大きいという問題がある. そこで, 前節の事前実験で得られた, 指示チューニング言語モデルは○年○月○日というフォーマットの回答が可能であるという知見を利用し, 該当部分を抽出したのちに exact match で評価を行う指標を採用する. 以下にその詳細を示す.

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{y}_i = y_i) \tag{1}$$

ここで, $\hat{y}_i$ は質問 q_i に対する回答 s_i から抽出された部分文字列, y_i は正解を表している. $\mathbb{I}$ は指示関数である. なお, モデルが年号と月日の情報を別々に保有している可能性を考慮して, accuracy は年月日 (date), 年 (year), 月日 (month and day 以下では md と省略する) の 3 つの軸で評価を行う.

[1] https://huggingface.co/google/gemma-2-2b-jpn-it
[2] https://huggingface.co/rinna/gemma-2-baku-2b-it
[3] https://huggingface.co/llm-jp
[4] https://huggingface.co/elyza/Llama-3-ELYZA-JP-8B
[5] https://huggingface.co/rinna/llama-3-youko-8b-instruct
[6] https://huggingface.co/cyberagent/calm3-22b-chat

図 1 各言語モデルの年代別の正答率を示したグラフ

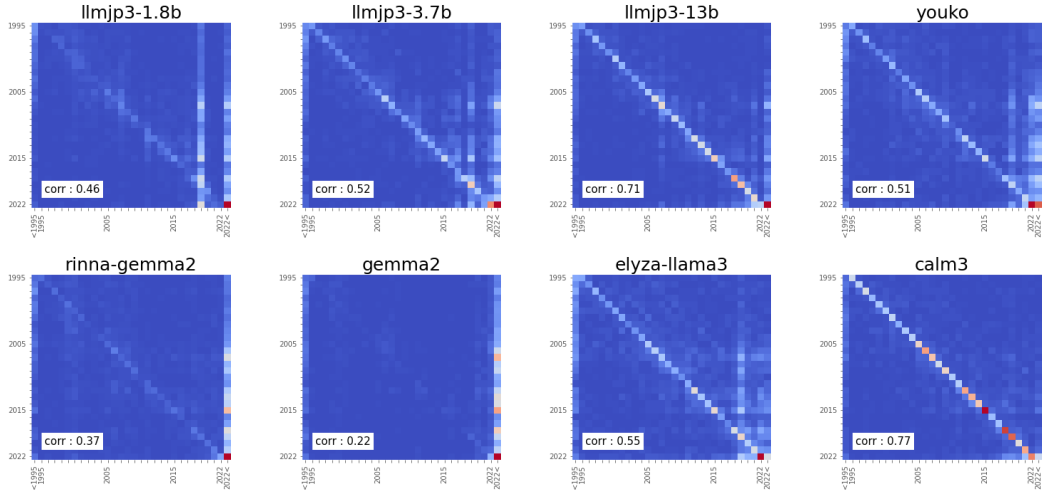

図 2 モデル出力に含まれる年号と解答の差異

5 実験結果

5.1 年代別の正解率

図 1 に, 各モデルの年代別の date, year, md の精度を示した. まず, 年代別の精度に着目すると, いずれの項目においても, 個々の年ごとに大きな差は見られない. 事前学習データに含まれる Web テキストの量は, 古い時代よりも新しい時代のイベントの方が相対的に多いと考えられる. このことから, より直近の事実に関する質問の正解率が高くなることが予想されるが, 今回の実験結果では, 年代と正解率の間に強い関係は見られなかった. モデル間の精度を比較すると, gemma2 や llmjp-1.8b のような 2B サイズのモデルは, 全て年代で精度が低い一方, llmjp-13b と calm3 は, 特に year で高い精度を示した. また, llmjp の 3 種類のパラメータサイズにおいても同様の傾向が見られ, 1.8b, 3.7b, 13b の順に正解率が向上している. この結果は, 年月日などの数字に関連する事実知識を正しく引き出すためには, モデルのパラメータサイズが重要な役割を果たすことを示唆している. 3 つの項目全体を見ると, どのモデルにおいても year の精度が最も高く, date と md の精度は比較的低い. これは, 言語モデルが事前学習と指示チューニングを通じて, イベントと月日を含めた時間的情報を正確に紐付けることができていないことを意味する.

5.2 実際の年号との乖離

図 2 に正解の年号とモデルが回答した年号の関係を混同行列のヒートマップとして示した. 例えば,

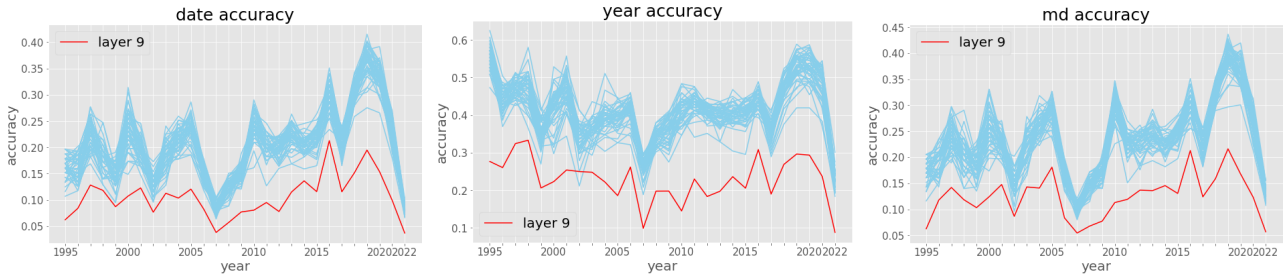

図 3 MLP モジュールを無効化した場合の正答率. 特に変化が顕著だった layer を赤, それ以外を水色で示している.

1995 年が正解となる質問に対してモデルが 1996 年と回答した場合, 縦軸 1995 年・横軸 1996 年の位置にカウントが記録される. このヒートマップにおいては, カウントの数に比例して色が赤くなり, モデルの正解率が高いほど対角線上のマスが赤色に近づく. モデルごとの比較を行った場合, llmjp-13b と calm3 が高い正解率を示していることがわかる. 一方, パラメータサイズが 2b と比較的小さい gemma2 は全ての年代でモデル出力が 2023 年に偏っており, イベントと発生年の対応関係を適切に学習できていない可能性が示唆される. この 2023 年への出力の偏りは llmjp シリーズでも確認された. 特に 2022 年のイベントについては, それ以降の年に予測が集中する傾向が見られる. モデルが年号 4 桁の最後の数字を予測する際, 自己回帰の結果 202 までたどり着いた場合, 理論上は 0〜9 の候補があるため, それぞれに結果が分散されることも考えられるが, 今回の実験では 2023 年への強い偏りが観察された. また, calm3 を除いたモデルに共通する特徴として, 予測結果がヒートマップの上三角領域に分布しているということが挙げられる. これは言語モデルが, イベントの発生時期を実際よりも後の年代として予測しやすい傾向があることを示唆している. ただし, 普遍的にこのような現象が存在するかどうかはモデルの追加も含めた更なる検討が必要である.

5.3 各層の MLP モジュールの重要性の調査

本節では, 言語モデルが保持する年月日の知識について, 内部モジュールに着目した分析を行う. 近年, 言語モデルの知識は attention ブロック内の MLP 層と密接な関係があるという説が有力となっている [8]. 特に, MLP 層の第 2 層は知識を key-value ストアの形式で保持しており, このモジュールのパラメータを更新することで知識の編集が可能であることが示されている [9]. 知識の特定方法については様々な手法が提案されている. attention の重みを観察する方法や, 勾配情報を利用する integrated gradient[8] はその代表的な例である. ただし, 前者は解釈可能な特徴を発見することが難しいこと, また, 後者については浅い層の勾配の影響が小さいという問題がある. causal tracing[9] は入力の埋込みを意図的に破損させた後に復元することで, 最終出力確率に影響を与えたトークンを特定する手法であるが, 計算コストが非常に高い. そこで, 本研究では, 先行研究でも採用されており [10][11], 尚且つ軽量である, モジュールの無効化を用いた知識の分析を行う. 無効化の対象モジュールは, 上記の先行研究に従い MLP 層に設定し, 第 0 層から最終層までを個別に無効化した際の出力を, 前節と同様の指標で評価する. 言語モデルは最も良い性能を示した calm3 を選択した. 図 3 に 48 層の attention ブロックをそれぞれ個別に無効化した際の, 年代別の精度を示した. date, year, md のいずれにおいても, 9 層目の MLP モジュールを無効化した場合の精度が著しく低下しており, その他の層はグラフの形状が類似していることが分かる. これは, 特定の層に年月日に関する知識がエンコードされていることを示唆するとともに, 言語モデルの知識は MLP モジュールと密接な関係がある, という先行研究の結果を支持するものである.

6 おわりに

本研究では, 言語モデルが有する時間に関する事実知識, 特に過去の出来事の日付に関する知識について, データセットの構築も含めた調査を行った. 実験の結果, 言語モデルは年号に関する知識については, 月日と比較してより正確に保持していることがわかった. また, 年号を正確に認識できていない言語モデルは, 実際の年よりも後の年代として推論しやすい傾向があることが示唆された. 今後は, 今回作成したデータセットの改良も含めて, 言語モデル内部の詳細な分析を行っていく予定である.

謝辞

本研究はJSPS科研費JP21K12036及びJP24K15093の助成を受けたものです。

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 30. Curran Associates, Inc., 2017.

[2] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, J. Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, P. Welinder, P. Christiano, J. Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. **ArXiv**, 2022.

[3] Siheng Xiong, Ali Payani, Ramana Kompella, and Faramarz Fekri. Large language models can learn temporal reasoning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 10452–10470, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[4] Zhao yu Su, Jun Zhang, Tong Zhu, Xiaoye Qu, Juntao Li, Min Zhang, and Yu Cheng. Timo: Towards better temporal reasoning for language models. **ArXiv**, Vol. abs/2406.14192, , 2024.

[5] Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. Timebench: A comprehensive evaluation of temporal reasoning abilities in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 1204–1228, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[6] 杉本智紀, 尾上康雅, 谷中瞳. アスペクトを考慮した日本語時間推論データセットの構築. 自然言語処理, Vol. 31, No. 2, pp. 637–679, 2024.

[7] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. A survey on llm-as-a-judge, 2024.

[8] 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.

[9] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. **Advances in Neural Information Processing Systems**, Vol. 36, , 2022. arXiv:2202.05262.

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

[11] Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed, 2024.

A Appendix

A.1 データセットの構築手順

A.1.1 Wikipedia からの情報抽出

上記のページには, 該当する年において日本で起きた出来事や流行・世相などの情報が, セクションごとに細かく記述されている. 年代別にデータセットを作成するためには, 特定のセクションについて, 全てのページでフォーマットが共通していることが条件となる. 1995 年〜2022 年のそれぞれのページを著者が目視で確認を行った結果, ”できごと”のセクションについて, 全ての年代のページで統一されていることが分かった. そこで, まず最初のステップとして, このセクションに記述されている情報から, イベント対日時のペアを抽出する. Wikipedia にはデータの書き出し機能が実装されており, 対象ページのテキストと編集履歴を XML 形式でダウンロードすることが可能である. 1995 年〜2022 年のページをこの機能を利用して手動で XLM ファイルに変換した後に, python ライブラリである xml.etree.ElementTree を用いて解析した. できごとのセクションはさらに月ごとのサブセクションに分かれており, 該当箇所を正規表現でマッチさせることにより, 日時とイベントの対を一括で取得した. 得られた文字列には wikilink のカッコや不要なタグが含まれており, これらについては事後処理で削除している.

A.1.2 イベントの分割

前節で取得したデータは基本的に, 特定のイベントのみを記述したものが大半だが, 中には図のように, 複数のイベントが混在しているパターンが見られた. このようなデータについては, 図のように個別のイベントに分割する必要がある. この操作をルールベースで記述することは困難であるため, ChatGPT を用いた実装を行った. 具体的には, Appendix の図に示したプロンプトを GPT-4o に入力し, 得られた応答を改行で split したものを, そのイベントが発生した日時と個別に紐付けた. つまり, { 日時: イベント } であったペアが, { 日時: イベント 1}, { 日時: イベント 2},... のように, 複数のペアに分割される. なお, イベントが抽出できない場合には, 「抽出不可」と出力するように指示をした.

A.1.3 QA 形式への変形

前節の処理を行った結果, { 日時: イベント } の形式のデータが得られた. この辞書型データからイベントの部分を利用して質問応答を作成する. 応答については, 日時そのものを回答として用いた. 文体の統一されていないテキストをルールベースで質問形式に変換することは難しいため, 前節同様, ChatGPT を用いることで作成を行った. 図に質問作成のためのプロンプトを示した. 質問の語尾形式を揃えるために, プロンプトでその旨を指示として組み込んでいる. また, 対象となるイベントと日時が一意に紐付くことが重要であるため, イベントテキスト中に含まれる固有名詞を可能な限り含めるように指示した. なお, こちらのタスクにおいても, 質問が作成不可の場合には「作成不可」と出力するように指示をした.

A.1.4 人手によるスクリーニング

本研究で作成するデータセットは, 実世界のイベントとその発生日時が紐づいた事実 QA データセットである. 作成にあたっては, Wikipedia に記載されている内容が事実であるという仮定を置いているが, ChatGPT で変換処理をした際に本来の事実が書き換えられたものが生成されてしまう可能性がある. 幻覚（hullcination）問題は, 言語モデルの性能が向上した現在においても, 解決が困難な問題の一つとされており, このような事態は避ける必要がある. また, 作成された全てのデータが有効なものであるとは限らない. そこで, 人手によるデータのスクリーニングを実行した. 具体的には, セクションとセクションで得られたデータをそれぞれ照らし合わせ, 以下の観点からチェックを行い, 条件にそぐわないものは除外, もしくは内容を修正した. 本作業は著者が行った.

- Wikipedia から抽出した事実がそのまま反映されているか
- 文法的に誤っている箇所が無いか
- その記述のみで一意に日時を特定できる質問になっているか