修士論文

Master's Thesis

論文題目

Thesis Title

言語モデルの学習における知識ニューロンの形成過程

On the Formation of Knowledge Neurons in Pre-trained Language Models

提出者

東北大学 大学院 情報科学研究科 システム情報科学専攻

Department of System Information Sciences

Graduate School of Information Sciences, Tohoku University

学籍番号 (ID No.) C2IM2002

氏名 (Name) 有山　知希 (Tomoki Ariyama)

指導教員	鈴木　潤　教授
学位論文指導教員	
審査委員 (〇印は主査)	〇鈴木　潤　教授　1 大林　武　教授 2 乾　健太郎　教授　3 坂口　慶祐　准教授

提出者略歴	
ありやま　ともき 氏名　有山　　知希	平成 10 年 5 月 31 日生
本籍　埼玉県	国籍
履歴事項	
[学歴]	
平成 30 年 4 月 1 日	東北大学　工学部　電気情報物理工学科入学
令和 4 年 3 月 25 日	同　卒業
令和 4 年 4 月 1 日	東北大学　情報科学研究科　システム情報科学専攻 博士課程前期 2 年の課程　入学
令和 6 年 3 月 25 日	同　修了

On the Formation of Knowledge Neurons in Pre-trained Language Models*

Tomoki Ariyama

Abstract

The language models used to solve tasks related to natural language processing are considered to have acquired the knowledge necessary to process the task through pre-training of the model. Various studies have investigated how such pre-trained language models store the knowledge acquired during training, and some of them report the existence of "knowledge neurons" that encode knowledge in the language model. In this study, we investigate how such knowledge neurons are formed in models over the course of pre-training. The results confirm the existence of neurons that satisfy the properties of knowledge neurons even in models in the untrained state, indicating that it is unlikely that knowledge neurons are formed over the course of pre-training. We then discuss the existence of ideal knowledge neurons in language models through various analyses of neurons found in models undergoing pre-training.

Keywords:

Natural Language Processing, Language Model, Pre-training, Knowledge Neuron, Interpretability

*Master's Thesis, System Information Sciences, Graduate School of Information Sciences, Tohoku University, C2IM2002, January 26, 2024.

言語モデルの学習における知識ニューロンの形成過程*

有山 知希

内容梗概

自然言語処理に関するタスクを解くために利用される言語モデルは，モデルの事前学習を通じてタスクの処理に必要な知識を獲得していると考えられる．このような事前学習済み言語モデルが，学習中に獲得した知識をどのようにモデル内に保存しているかについては様々な研究が行われているが，それらの中には言語モデル内に知識をエンコードしている “知識ニューロン” の存在を報告しているものがある．本研究では，そのような知識ニューロンが事前学習の経過に伴ってどのようにモデル内に形成されるかを調査する．調査の結果，学習が進んでいない状態のモデル内にも知識ニューロンの性質を満たすようなニューロンが存在することが確認され，知識ニューロンが事前学習の経過に伴って形成されている可能性が低いことが示された．その結果を踏まえた上で，事前学習中のモデルから発見されるニューロンに関する様々な分析を通じ，言語モデルにおける理想的な知識ニューロンの存在についての議論を行う．

キーワード

自然言語処理，言語モデル，事前学習，知識ニューロン，解釈可能性

*東北大学 大学院情報科学研究科 システム情報科学専攻 修士論文, C2IM2002, 2024 年 1 月 26 日.

目 次

図 目 次

表 目 次

1　はじめに

言語モデルは自然言語処理に関するタスクを解くために利用されており，近年では言語モデルを作成する方法として，ニューラルネットワークを用いた深層学習モデルに膨大な量のテキストを学習させることが広く行われている．テキストの学習方法には様々な種類が存在するが，事前に学習を行った言語モデルの中には，“___はニャーと鳴く．” という穴埋め文が与えられた時に，穴埋め部分に “猫” が入ると予測できるものがある．猫についての知識がなければこの穴埋め部分を正しく予測することはできないため，このような言語モデルには，事前学習によって何らかの形で猫に関する知識が保存されていると考えられる．事前学習済み言語モデルにおける知識については，Dai ら [1] や有山ら [2] によって，Transformer[3] モデルの Feed-Forward 層に知識をエンコードしていると考えられるニューロン，すなわち “知識ニューロン” が存在することが報告されている．

また，ChatGPT のようなチャットボットなどに代表されるように言語モデルは実社会の中で普及しつつある段階にあるが，なぜ自然な応答が返せるのかといった，自然言語処理タスクを解くことができる仕組みをモデルの内部から説明することについては，依然として達成できていない事が多い状態である．そのため，現状では言語モデルはブラックボックス性が高い状態にあり，モデルに対する信頼性という観点からも，言語モデルの説明可能性を向上させる研究の需要が高まっている．

こうした背景を踏まえ，本研究では知識ニューロンが事前学習中にどのように言語モデルに形成されていくかを調査することにした（図 1）．学習過程における調査を行うため，学習途中のモデルが利用可能である MultiBERTs[4] と Pythia[5] の各チェックポイントに対し，Dai ら [1] によって提案された知識ニューロンを探す手法を適用した．その結果を学習経過に沿って分析することで，知識ニューロンの形成過程を調査した．

調査の結果，学習途中のモデル内にも知識ニューロンの性質を満たすニューロンが発見され，これまで知識ニューロンであると考えられていたものが，実際には理想的な意味での “知識ニューロン” の性質を満たしていない可能性を示した．また，それらの結果の分析を通じて，言語モデルにおける理想的な知識ニューロ

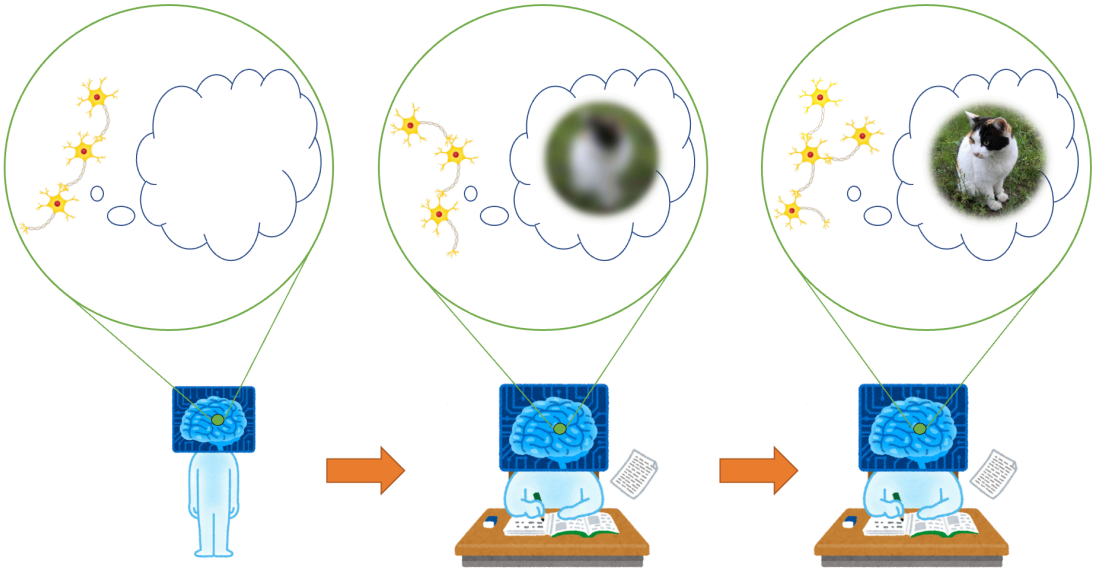

図 1: 知識ニューロンは学習によってどのように形成されていくのか？

ンについての議論を行った.

2 関連研究

2.1 言語的特性とモデル内部表現

“どのような言語的特性がモデル内部で用いられる表現に学習されているのか”という疑問に対する研究は，深層学習言語モデルの振る舞い・性質の理解に向けて盛んに行われている．例えば，“埋め込み” と呼ばれるベクトル表現については，文の長さや単語の内容，語順などの特性を捉えていること [6] や，BERT[7] モデルでは，品詞タグのような基本的で構文的な情報はモデルの入力に近い層で，共参照のような高レベルで意味的な情報は出力に近い層で処理されている [8] などの結果が報告されている．本研究も，言語的知識がモデル内部に獲得される過程を調査することを目的としているため，これらの研究の一環に位置付けられる．

2.2　ニューロンを用いた分析

2.1 節で述べたような，埋め込みやモデルを構成する層レベルでの分析を行う研究が行われている一方で，より細かいレベルである “ニューロン” という観点からモデルを解釈しようとする研究が注目され始めている．ニューロンという用語は，Transformer モデルのようなニューラルネットワークを構成する要素のうち，単一次元の出力を指すために使われることが一般的である．このニューロンレベルでの分析を行っている研究としては，特定の n-gram や入力における位置の情報に反応するニューロンの存在を報告しているもの [9] や，GPT-2[10] で活性化しているニューロンの説明を GPT-4[11] によって生成することで，言語モデルの解釈可能性を言語モデルによって高めることを目指したもの [12] などがある．

本研究では知識ニューロンの形成過程を調査するが，その中で使用する知識ニューロンを探すための手法 [1] もこれらの研究の一端である．本研究の特徴は，知識ニューロンの存在を報告することを目的としている訳ではなく，それらがどのようにモデル内で形成されていくかということに焦点を当て，そのテーマに人間が解釈可能な形で説明を与えたことにある．

2.3　言語モデルが持つ知識の編集

深層学習技術の進歩によって言語モデルの出力は自然さを増しているが，その出力に事実と異なる内容が含まれていることがあり，実社会における言語モデルの普及を妨げている一因となっている [13]．また，近年の深層学習言語モデルには莫大なパラメータサイズを持つものが多く登場している．パラメータサイズが大きくなるにつれて学習やファインチューニングにかかる計算資源などのコストも大きくなるため，モデルが誤った事実を学習してしまった場合に，それらをファインチューニング等によって更新しようとすると，膨大なコストと時間がかかってしまう．さらに，モデルから誤っている事実のみを修正する方法は依然として確立されていないのが現状である [14]．

そのため，モデル内のどのパラメータに言語的知識が構造化されているかを特定し，そのパラメータを更新することによって誤った事実のみを低コストに更新

する手法の研究が行われている [1, 15, 16, 17, 18]．本研究は，言語モデルが知識を学習する過程の一端を明らかにしているという観点から，誤った知識を獲得してしまう過程に示唆を与える可能性がある．

3　実験手法

3.1　Transformer モデルにおけるニューロン

実験手順を説明する前に，本論文において重要な用語である “ニューロン” について説明を行う．本論文におけるニューロンとは，Transformer[3] を構成する 1 モジュールである Feed-Forward 層（以下，“FF 層” と呼ぶ）において，第一線形層の出力を活性化関数にかけたものを指す（図 2 も参照）．ここで，FF 層の式は入力を $\boldsymbol{x}$，第一・第二線形層の重み行列・バイアス項をそれぞれ $\boldsymbol{W}_1$, $\boldsymbol{b}_1$, $\boldsymbol{W}_2$, $\boldsymbol{b}_2$ で表し，活性化関数に GELU[19] と呼ばれる関数を用いると，次の式 (1) の形で表される：

$$\mathrm{FF}(\boldsymbol{x}) = (\mathrm{GELU}(\boldsymbol{x}\boldsymbol{W}_1 + \boldsymbol{b}_1))\boldsymbol{W}_2 + \boldsymbol{b}_2 \tag{1}$$

式（1）中の “$\mathrm{GELU}(\boldsymbol{x}\boldsymbol{W}_1 + \boldsymbol{b}_1)$” の部分がニューロンの行列に対応し，各次元の値がニューロンの活性値となる．

このニューロンの定義は，Geva ら [20] によって FF 層が key-value メモリと呼ばれる機構と同様の働きをすることが報告されていることに基づいている．Key-value メモリとは，キー行列 $\boldsymbol{K}$ とバリュー行列 $\boldsymbol{V}$ によって，入力 $\boldsymbol{x}$ を期待される値に変換する機構のことであり，数式で表現すると次の式 (2) のようになる．

$$\text{Key-value memory}(\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{x}\boldsymbol{K}) \cdot \boldsymbol{V} \tag{2}$$

この式 (2) は式 (1) とほとんど同一であり，活性化関数の種類とバイアス項の有無だけが異なる．そのため，FF 層はモデルが学習中に獲得した知識を保存している可能性が高いと考え，FF 層の中間表現の各次元をニューロンとして扱うことにした．

3.2 知識ニューロンを探すためのタスク

3.2.1 マスク言語モデルに用いるタスク

マスク言語モデルとは，一部分が隠された文（=穴埋め文）が学習データとして与えられ，隠された部分に元々何という語が入っていたかを予測するように学習されたモデルのことである．そのため，知識ニューロンを探すためのタスクとして，マスク言語モデルに対しては穴埋め文の穴埋め部分を予測させるタスクを使用する．ここで穴埋め文は，穴埋め文中に登場する概念[1]を隠すように，すなわち [MASK] トークンに置き換えるようにすることでデータセット（4.1.2 節）から作成する．

また，今後の説明のために穴埋め文および概念の呼び方を定義する．今，ある 1 つの概念 C を考えているとする．このとき，C が穴埋め部分に入る穴埋め文を，概念 C の「正例文」と呼び，概念 C そのものを「正例概念」と呼ぶ.. 対して，C が穴埋め部分に入らないものは「負例文」と呼び，C 以外の概念を「負例概念」と呼ぶことにする．例えば，下記は概念として “cat” を扱う時に考えられる正例文と負例文の例であり，正例概念は “cat”，負例概念は “become” である：

- 正例文: A kitten will grow and become a [MASK].
- 負例文: A kitten will grow and [MASK] a cat.

3.2.2 生成型言語モデルに用いるタスク

生成型言語モデルとは，与えられた文の後ろに続くトークンを予測するように学習されたモデルのことである．そのため，生成型言語モデルに対しては，知識ニューロンを探すためのタスクとして，途中で途切れている文の次の単語を予測させるタスクを使用する．ここで，途中で途切れている文は次に続く単語に概念が来るようにするため，元々概念を文中に含む文に対し，その概念以降の部分を削除することでデータセット（4.1.2 節）から作成する．

[1]本論文において「概念」とは，名詞や固有名詞等で表されるエンティティや，動詞や形容詞等で表される動作や性質などを表す，あらゆる単語として定義する．

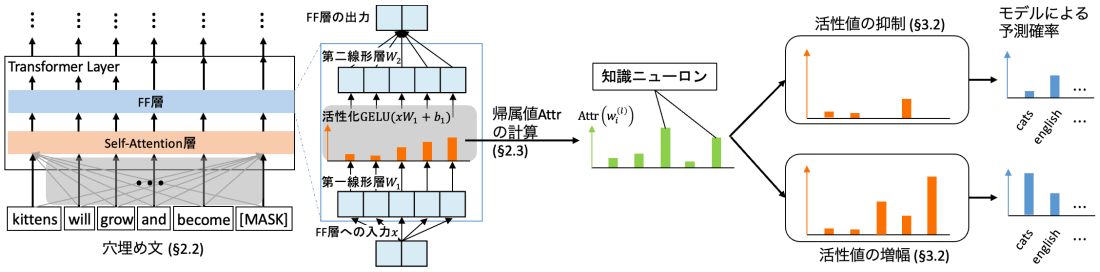

図 2: マスク言語モデルの場合の知識帰属法と活性値編集のイメージ．知識帰属法では，モデルが穴埋め文の穴埋め部分を予測する際の，FF 層における各ニューロンの活性値を用いて帰属値を計算し，それらを元に知識ニューロンを探し出す．

穴埋め文の時と同様に，途中で途切れている文についても正例文と負例文の呼び方を導入する．ある 1 つの概念 C について，次に続く単語に C が入る場合を正例文，入らない場合を負例文と呼ぶ．以下は，概念 “cat” の正例文と負例文の例である：

- 正例文: A kitten will grow and become a
- 負例文: A puppy will grow and become a

3.3 知識帰属法

本節では，Dai ら [1] によって提案された，事前学習済み言語モデルから知識ニューロンを探す手法である知識帰属法について説明する (図 2 参照)．以下では，ある 1 つの概念 C を扱うことを考え，その概念 C についての知識をエンコードしていると考えられる知識ニューロンを探す方法について述べる．

まず，言語モデル内に存在する各ニューロンのうち，“モデルが，概念 C の正例文 s に対して正しい答え C を出力する確率 $\mathrm{P}_s(\hat{w}_i^{(l)})$” に大きく影響を与えるものを探す．影響の大きさは帰属値 $\mathrm{Attr}(w_i^{(l)})$ によって測るため，その計算方法を説明する．

帰属値 $\mathrm{Attr}(w_i^{(l)})$ の計算に必要な上述の確率 $\mathrm{P}_s(\hat{w}_i^{(l)})$ は，$w_i^{(l)}$ を l 番目の FF 層の i 番目のニューロン，$\hat{w}_i^{(l)}$ をそのニューロンの活性値とすると，次の式（3）で

与えられる：

$$\mathrm{P}_s(\hat{w}_i^{(l)}) = p(C|w_i^{(l)} = \hat{w}_i^{(l)}) \tag{3}$$

この確率について，Sundararajan ら [21] の “Integrated Gradients” という帰属法を用い，$\hat{w}_i^{(l)}$ を 0 から事前学習済み言語モデルにおける活性値 $\bar{w}_i^{(l)}$ まで変化させたときに，それに伴って変化する，確率 $\mathrm{P}_s(\hat{w}_i^{(l)})$ に対するニューロン $w_i^{(l)}$ の勾配 $\frac{\partial \mathrm{P}_s(\hat{w}_i^{(l)})}{\partial w_i^{(l)}}$ を積分することで，帰属値 $\mathrm{Attr}(w_i^{(l)})$ が計算される：

$$\mathrm{Attr}(w_i^{(l)}) = \bar{w}_i^{(l)} \int_{\hat{w}_i^{(l)}=0}^{\bar{w}_i^{(l)}} \frac{\partial \mathrm{P}_s(\hat{w}_i^{(l)})}{\partial w_i^{(l)}} d\,\hat{w}_i^{(l)} \tag{4}$$

この値が大きいほど，正例文 s に強く反応するニューロンであると判断する．この方法を用いてモデル内の全てのニューロンの s に対する帰属値を計算し，その中から帰属値の閾値 t を超えるニューロンのみを選ぶことで，正例文 s に強く反応するニューロンを選出することができる．

しかし，このように 1 つの正例文 s に反応するニューロンを選び出しても，それらは必ずしも概念 C の知識ニューロンであるとは限らない．なぜなら，式（4）の帰属値はニューロンが正例文 s に反応する度合いを表す値のため，s 内の他の単語や構文情報などに反応しているような「偽陽性の」ニューロンが選ばれているかもしれないからである[2]．

そこで，先述した帰属値の閾値によるニューロンの選出を複数の正例文に適用する方法を採ることで，C の知識ニューロンを発見することができる：

1. 概念 C の正例文を，構文や含まれる語彙が異なるようにして複数用意する
2. 各正例文について，モデル内の全てのニューロンの帰属値を計算する
3. 各正例文について，閾値 t を超える帰属値を持つニューロンのみを選出する
4. 全ての正例文間での共有率の閾値 p を設定し，全ての正例文のうち p% 以上で選出されているニューロンのみを残す

[2]例えば，“A kitten will grow and become a [MASK].” という穴埋め文に反応するニューロンがいくつか選出されたとしたとき，それらの中には単語 “grow” や “A and B” という構文情報に反応しているニューロンが含まれている可能性がある．

最後のステップ 4. で残ったニューロンは，各正例文で共有されている要素，すなわち “概念 C” の知識ニューロンである．

なお，ここで注意として，概念 C の知識ニューロンとは C の知識をエンコードしているニューロンのことを指すため，パラメータがランダム初期化のモデルや学習ステップ数が不十分なモデルに知識帰属法を適用して発見されるニューロンは，その学習の不十分さゆえに “知識ニューロン” とは呼べない．そのため，以下では知識帰属法によって発見されるニューロンのことを，それが知識ニューロンである場合も含め，“特定の概念の予測に寄与するニューロン” という意味で “寄与ニューロン” と呼ぶ．

4 実験: 知識ニューロンの形成過程

4.1 設定

4.1.1 モデル

本論文では実験対象の事前学習済み言語モデルとして，マスク言語モデルと生成型言語モデルを用いる．

まず，マスク言語モデルとしては MultiBERTs[4] を用いる．この MultiBERTs は事前学習過程に関する研究の促進を目的として，オリジナルの BERT[7] モデルと同様のハイパーパラメータで訓練された BERT-base の，学習途中の状態のモデル（“チェックポイント” と呼ばれる）が公開されているモデルである．本論文では，公開されている MultiBERTs のうち seed 値 0 のものを用いた．なお，seed 値 0 の MultiBERTs では学習ステップ数，すなわちパラメータの更新回数が 0（=ランダム初期化状態のモデル）及び 20k から 200k まで 20k 毎のチェックポイント，200k から 2000k まで 100k 毎のチェックポイントの計 24 個のチェックポイントが公開されているが，可視化の都合から以下に記載する 13 個のチェックポイントに対する実験結果を報告する．

- 0, 20k, 40k, 60k, 80k, 100k, 200k, 300k, 400k, 500k, 1000k, 1500k, 2000k

表 1: 実験に使用した各モデルのパラメータの詳細.

	パラメータ総数	層数	隠れ層の次元	ニューロン総数
MultiBERTs	110M	12	768	36,864
Pythia	70M	6	512	12,288
	160M	12	768	36,864
	410M	24	1,024	98,304

続いて生成型言語モデルについては，Pythia[5] を用いる．この Pythia も，学習段階における言語モデルの発展・進化の過程を調査することを目的として公開されている GPT-3[22] ライクなモデルである．さらに Pythia は，モデルパラメータの数の大小が，モデルの学習における発展・進化に与える影響についても調査することを目的として掲げており，そのため小さいものは 70M パラメータから大きいものは 12B パラメータを持つモデルが公開されている．本論文では，計算資源の制約と実装の要因から，以下の 3 つのパラメータサイズを持つモデルに対して実験を行う．なお “-deduped” とは，繰り返されるデータの重複を排除した学習データを用いて学習されたことを表す．本実験では，そのように構成された学習データを用いたモデルの方がそうでない学習データを用いたモデルと比べて，同じ学習ステップ数でもより効率の良い学習が可能で多くの知識を保持している可能性があると考え，“-deduped” バージョンのモデルを実験対象として採用した．

- 70M-deduped, 160M-deduped, 410M-deduped

また各パラメータサイズのモデルには，学習ステップ数が 0（=ランダム初期化状態のモデル）と 1 から 512 まで 2 の冪乗毎のチェックポイント，及び 1k から 143k まで $1k$ 毎のチェックポイントの計 154 個のチェックポイントがそれぞれ公開されているが，可視化の都合から以下に記載する 13 個のチェックポイントに対する実験結果を報告する[3]．

[3]学習ステップ数はあくまで “モデルパラメータの更新回数” であるため，MultiBERTs と Pythia の間で学習ステップ数の比較をすることは学習の量という観点からは意味がないことに留意する．すなわち，学習ステップ数が同じでも用いる学習データやバッチサイズが異なれば，一度のパラメータ更新でモデルが見るデータの量は異なる．

- 0, 512, 1k, 3k, 5k, 10k, 20k, 40k, 60k, 80k, 100k, 120k, 143k

実験で使用する各モデルの詳細については，表 1 に示す．

4.1.2 データセット

マスク言語モデルのタスクの際に用いる穴埋め文については，Generics KB[23] を用いて作成した．このデータセットは自然かつ意味的に正しい文を大規模に提供しているだけでなく，その文のトピックである単語の情報も含んでいることから，その単語を概念として扱うことで簡単に大量の穴埋め文を作成できるため使用した．また，Generics KB にはいくつかのデータセットセクションが提供されているが，中でも自然な文が提供されている GenericsKB-Best のデータを用いて概念部分をマスクして作成した．実験に使用した概念は，3.3 節で述べた “全ての正例文のうち 50%以上で選出されているニューロンのみを残す” というステップを踏めるよう，GenericsKB-Best データセットから正例文が 4 つ以上作成できたものに限定し，最終的に 4,207 個の概念について穴埋め文を作成した．

生成型言語モデルのタスクの際に用いる文については，Natural Questions[24] を用いて作成した．このデータセットは質問応答のために作られたものであるが，“次単語に概念が来る” という制約の文を作るという目的において，質問応答データセットにおける質問文は答えが一意に定まる性質を持つため，その答えを概念として扱うことで簡単に所望の文を手に入れることができる．このような理由から，本実験では質問応答データセットとして用いられることが多い Natural Questions を使用することにした．なお，Natural Questions の質問文をそのまま用いるだけでは，各概念について正例文を 4 つ以上作成することが難しかったため，TextAttack[25] という手法を用いて質問文の一部をパラフレーズすることによって正例文の数を確保した．パラフレーズの割合は元の質問文における単語のうちの 20% とし，この工夫によって最終的に 1,406 個の概念について 4 つ以上の正例文を作成することに成功した．

それぞれのデータセットから作成した文の例を，表 2 に示す．

表 2: 作成したデータセットの具体例．“##” が付くトークンは，直前のトークンに続いて単語を形成することを表す．

	正例概念	正例文
MultiBERTs	baseball	[MASK] is considered the national sport of cuba .
	thunder	[MASK] rolls from cloud to cloud .
	grammar	[MASK] ##s usually classify verbs as regular and irregular .
Pythia	heart	pace maker is associated with which body organ
	white	what color is the cue ball in pool
	France	what’s the biggest country in western europe

4.1.3 ハイパーパラメータとその他の設定

Dai[1] らの設定を参考に，3.3 節の知識帰属法で用いる，各正例文における帰属値の閾値 t は各正例文について得られた最も大きい帰属値の 0.2 倍とし，全正例文間の共有率 p は 50% に設定した．

4.2 事前学習における知識ニューロンの形成過程

本節では，学習の経過による知識ニューロンの形成過程を調べるための実験手順について説明する．

知識ニューロンの形成過程を調査するためには，モデルの学習ステップ数の増加に伴う，寄与ニューロンの “知識ニューロンらしさ” の変化を調べれば良い．すなわち，学習が進むに従って寄与ニューロンが “知識ニューロンらしさ” を増していく様子を観察することができれば，知識ニューロンが形成される過程を調べることができる．ここで “知識ニューロンらしさ” を，先行研究である Dai ら [1] に則って以下のように定義する：

(i) モデル内で知識ニューロンを抑制した場合，正例文を正解する確率は減少する

(ii) 一方で，同様の抑制操作を行っても，負例文を正解する確率は変化しない

言い換えると，“正例概念の出力のみに影響すること” を知識ニューロンの持つ性質として考える．この定義を踏まえ，次のような手順で実験を行う：

1. 1 つのチェックポイントに知識帰属法を適用し，各概念の寄与ニューロンを見つける
2. そのチェックポイントに正例文と負例文を予測させ，それぞれ正解する確率を測定する
3. そのチェックポイントの寄与ニューロンの活性値を抑制（後述）した上で正例文と負例文を予測させ，正解する確率を測定する (図 2 参照)
4. 手順 2. と 3. で測定した正例文についての正解確率の相対変化率（後述），および負例文についての正解確率の相対変化率を計算することで，そのチェックポイントで発見される寄与ニューロンが先述の “知識ニューロンらしさ” をどの程度持っているかを観察する
5. 手順 1. から 4. を，MultiBERTs や Pythia の各チェックポイントに対して行う

ここで，活性値の抑制方法については，次の 2 通りを検証する：

- 活性値を 0 に置き換えることで抑制
- 活性値を，他の適当な入力[4]をした際の活性値に置き換えることで抑制

前者は “エンコードされている知識を削除する” という操作の直感に従って値を 0 にしている．これは，先行研究の Dai ら [1] において使われていた方法であり，ここでもこれに倣う．後者は 0 という値が，通常の入力によって活性値が取りうる値の範囲外になっている可能性を想定している．これは，仮にあるニューロンの活性値を 0 にしたことでモデルが通常ではなり得ない状態になってしまった場合，そのようなモデルに対する実験を行っても，得られる結果には意味がなくなってしまうことを考慮している．

[4]具体的にどのような入力を使用したかについては，付録 A 章に示す．

相対変化率の計算方法については，寄与ニューロンの活性値を抑制する場合，正例・負例文共通で次の式 (5) によって計算される：

$$\frac{(\text{抑制後の正解確率} - \text{抑制前の正解確率}) \times 100}{\text{抑制前の正解確率}} \tag{5}$$

なお，この一連の実験の内，手順 3. における寄与ニューロンの活性値操作を “2 倍に増幅” (図 2 参照) に変更した場合の実験も行う．この場合は，定義 (i) が “知識ニューロンを強化した場合，正例文を正解する確率は増加する” という内容に変わった上で，知識ニューロンらしさの度合いを手順 4. で観察する．相対変化率の計算についても，寄与ニューロンの活性値を増幅する場合は，式 (5) 中の “抑制” を “増幅” に変更して計算する．

5　実験結果

5.1　マスク言語モデル

4.2 節の実験により得られた結果の内，MultiBERTs モデルについて，寄与ニューロンの活性値を 0 にすることで抑制した際の結果を図 3 に，他の適当な入力をした際の活性値に置き換えることで抑制した際の結果を図 4 に，増幅した際の結果を図 5 に示す．なお，プロットはその相対変化率を記録した概念の個数を表している．

まず，先行研究で提案されているニューロン抑制方法である，活性値を 0 にする方法を適用した際の結果である図 3 に着目する．図 3 において，学習ステップ数が最も大きい 2000k のチェックポイントの結果を見ると，抑制操作の前後で負例文予測の正解確率の相対変化率は 0% に近い値を記録している概念がほとんどである．一方で，正例文予測の際はその相対変化率が下がっている概念が多く，−50%から −75%の間の値を記録している概念が最も多い．これらのことは，学習が進んだ状態のモデルでは，ある概念について見つかる寄与ニューロンはその概念の出力確率のみを下げる傾向があり，その他の概念の出力確率には影響を与えない傾向があることを示している．すなわち，学習が進んだ状態のモデルに対して見つかる寄与ニューロンは，知識ニューロンらしさを持っていると言える．

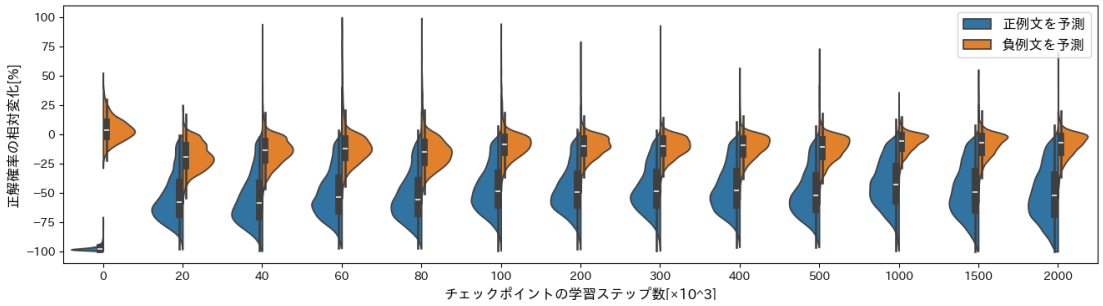

図 3: MultiBERTs について，各概念の寄与ニューロンの活性値を 0 に抑制した際の，穴埋めを正解する確率の相対変化率を，モデルの学習ステップ数ごとに示したもの．

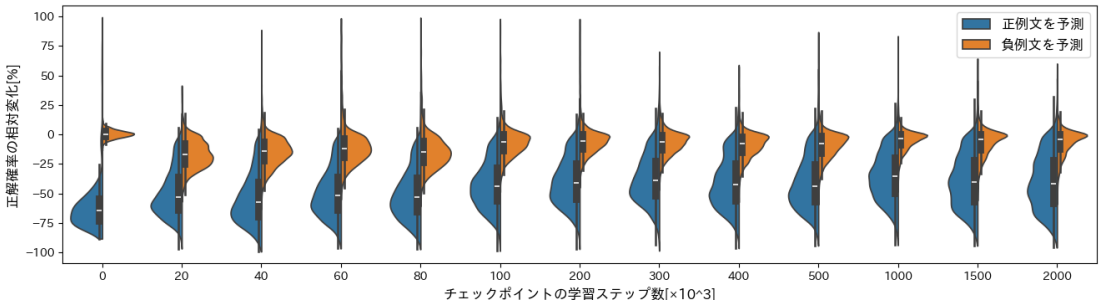

図 4: MultiBERTs について，各概念の寄与ニューロンの活性値を，適当な入力をした際の活性値に置き換えることで抑制した際の，穴埋めを正解する確率の相対変化率を，モデルの学習ステップ数ごとに示したもの．

しかし，この傾向は学習途中のチェックポイントに対しても見受けられることが図 3 から分かる．なぜならば，学習ステップ数が 20k から 1500k のチェックポイントにおけるグラフの概形も，2000k のものとほとんど同じだからである[5]．このことは，4.2 節で定義した “知識ニューロンらしさ” は事前学習の過程によって獲得されているのではなく，MultiBERTs のモデルにはある概念の出力のみに影響を与えるようなニューロンが学習過程に関わらず存在している，ということを示している．

次に，もう一つのニューロン抑制方法である，活性値を他の適当な入力をした

[5]学習ステップ数 0k の結果については，6 章で分析する．

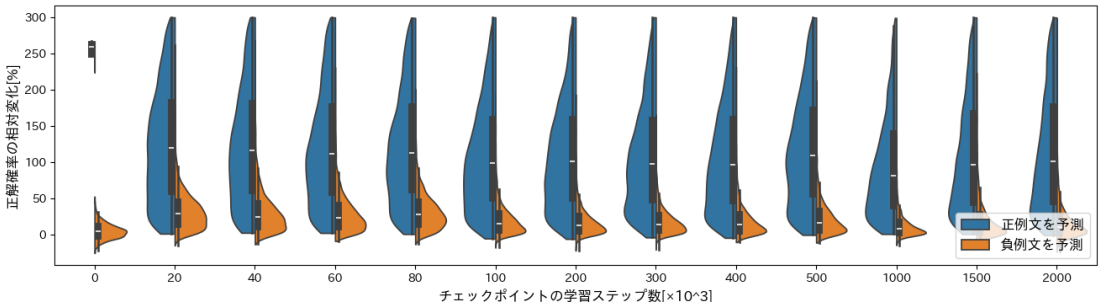

図 5: MultiBERTs について，各概念の寄与ニューロンの活性値を 2 倍に増幅した際の，穴埋めを正解する確率の相対変化率を，モデルの学習ステップ数ごとに示したもの．

際の活性値に置き換えて抑制した場合の結果である図 4 を見ても，学習過程を通じてグラフの概形は似通った結果となっていることが確認でき，活性値を 0 にすることで抑制した際と同じ結論を示唆する結果となっている．

次に，寄与ニューロンの活性値を 2 倍に増幅する操作を行った結果である図 5 について報告する．知識ニューロンの活性値を増幅すると，正例概念の出力確率が上昇することが有山ら [2] によって報告されており，その上昇率は，活性値編集前のモデルにおける出力確率によっては +100%を超えることがある．本実験でも同様の現象が観察されたが，中には外れ値的に +3000%を超えるような大きな上昇率を記録するケースが観察された．そのようなケースを含めてしまうとグラフの視認性が損なわれ，グラフを見ることで学習過程における傾向を調査することが難しくなってしまったため，活性値増幅操作の結果である図 5 にはグラフの視認性を損なわない範囲として，相対変化率が +300%を超えるケースは除いたものを記載している．その上で図 5 を見ると，4.2 節での増幅操作の場合の “知識ニューロンらしさ” の定義を満たすようなニューロンが，やはり学習過程に関わらず存在していることが確認される．このことから，活性値増幅操作による結果からも，寄与ニューロンが学習過程によって “知識ニューロンらしさ” を獲得しているわけではないと言える．

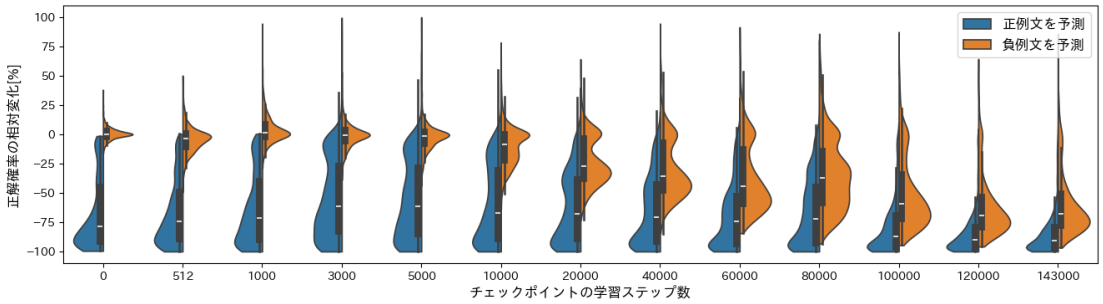

図 6: Pythia-70M について，各概念の寄与ニューロンの活性値を 0 に抑制した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

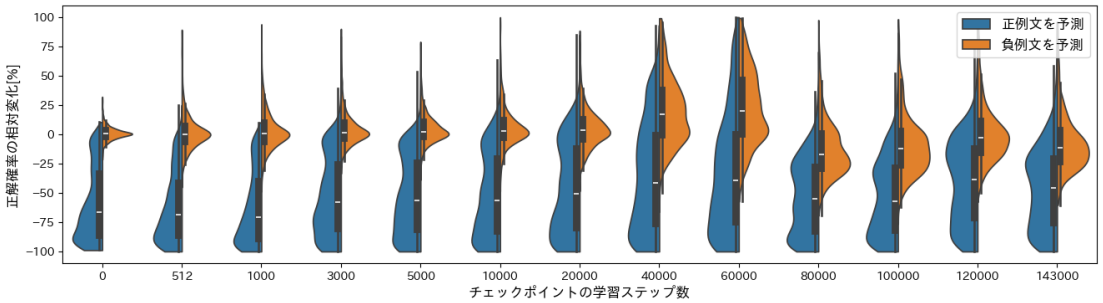

図 7: Pythia-70M について，各概念の寄与ニューロンの活性値を，適当な入力をした際の活性値に置き換えることで抑制した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

5.2　生成型言語モデル

本節では 4.2 節で示した実験手順を，生成型言語モデルである Pythia の各パラメータサイズに対して行った結果を報告する．

まず，図 6 と図 7 に，70M パラメータの Pythia モデルに対して活性値抑制操作を行った際の結果を，図 8 に活性値増幅操作を行った際の結果を示す．図 6 と図 7 の学習ステップ数 0 から 5000 までのグラフに着目すると，抑制操作によって正例文予測時の正解確率は相対的に減少する一方で，負例文予測時の正解確率はほとんど変化していない．この結果は，これらのチェックポイントに知識帰属法を適用して発見される寄与ニューロンが “知識ニューロンらしさ” を持っている

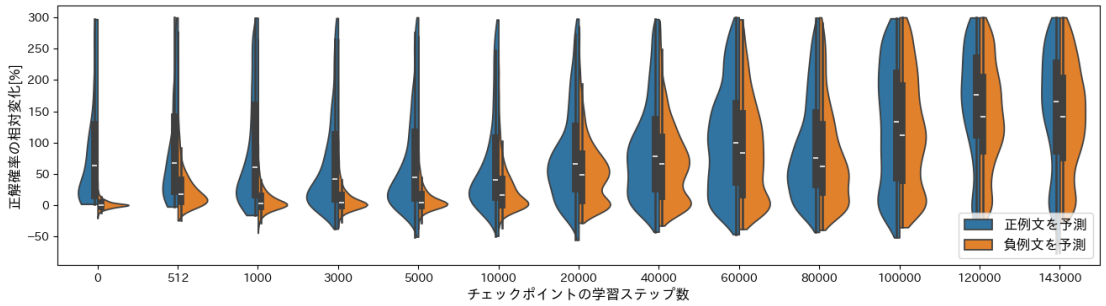

図 8: Pythia-70M について，各概念の寄与ニューロンの活性値を 2 倍に増幅した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

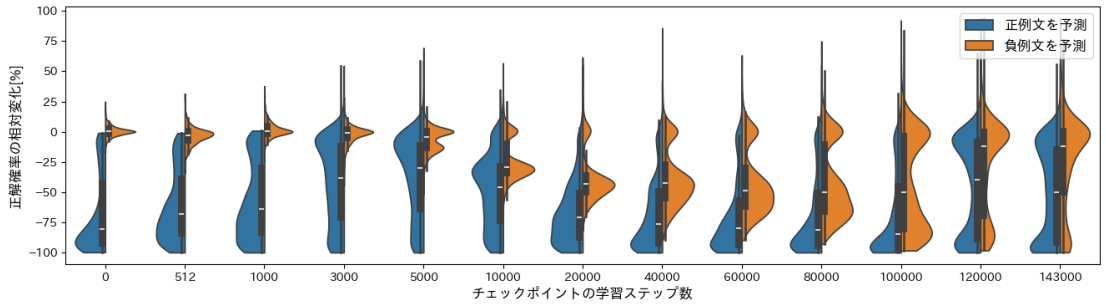

図 9: Pythia-160M について，各概念の寄与ニューロンの活性値を 0 に抑制した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

ことを示している．一方で学習ステップ数 $100,000$ から $143,000$ のグラフに着目すると，特に図 6 では正例文・負例文のどちらの場合も正解確率が減少しており，“知識ニューロンらしさ” は確認できない．また，活性値増幅操作を行った結果である図 8 に着目しても，抑制操作時と同様に，学習ステップ数の少ない段階のチェックポイントで知識ニューロンらしさを持つ寄与ニューロンが観察され，学習ステップ数の多い段階のチェックポイントでは知識ニューロンらしさが見られない結果となっている．以上の結果より，Pythia の 70M パラメータモデルでは，寄与ニューロンが “知識ニューロンらしさ” を獲得している様子を観察することはできなかった．

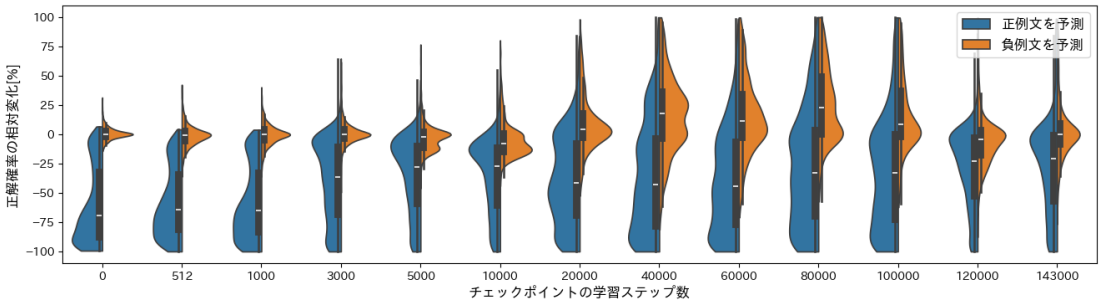

図 10: Pythia-160M について，各概念の寄与ニューロンの活性値を，適当な入力をした際の活性値に置き換えることで抑制した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

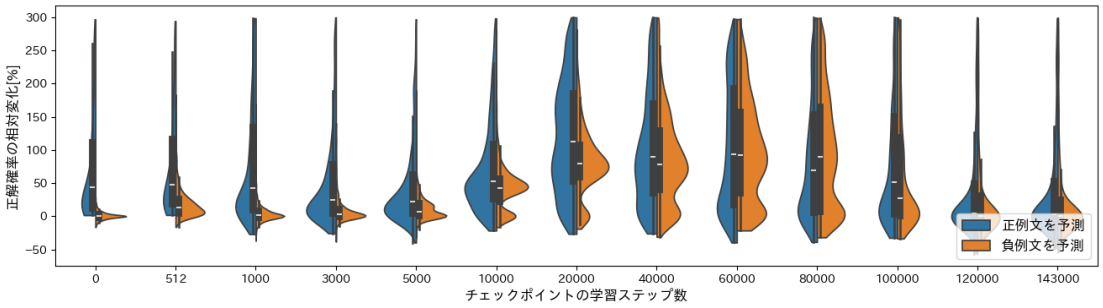

図 11: Pythia-160M について，各概念の寄与ニューロンの活性値を 2 倍に増幅した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

続いて，図 9 と図 10 に，160M パラメータの Pythia モデルに対して活性値抑制操作を行った際の結果を，図 11 に活性値増幅操作を行った際の結果を示す．同様にして，図 12 と図 13 には 410M パラメータの Pythia モデルに活性値抑制操作を行った際の結果を，図 14 には活性値増幅操作を行った際の結果を示す．全ての図に共通していることは，学習ステップ数が少ないチェックポイントで見つかる寄与ニューロンには知識ニューロンらしさが観察される一方で，学習ステップ数が多いチェックポイントで見つかるものには知識ニューロンらしさを持つ傾向が見られない，ということである．そのため，70M パラメータモデルについての結果も含めて，本研究で用いた生成型言語モデルでは，そもそも学習済み段階に

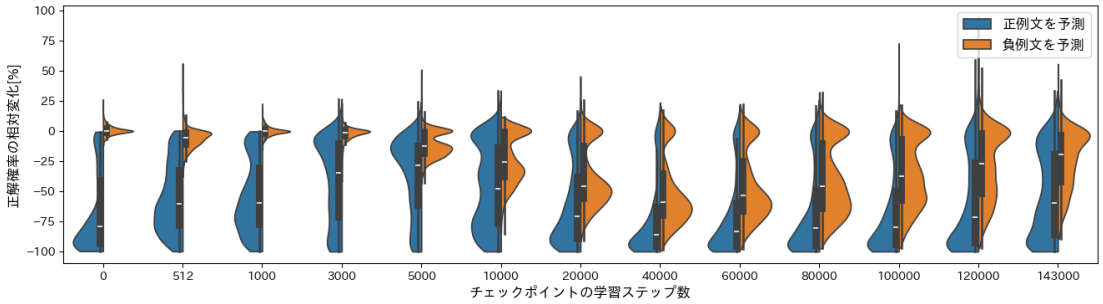

図 12: Pythia-410M について，各概念の寄与ニューロンの活性値を 0 に抑制した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

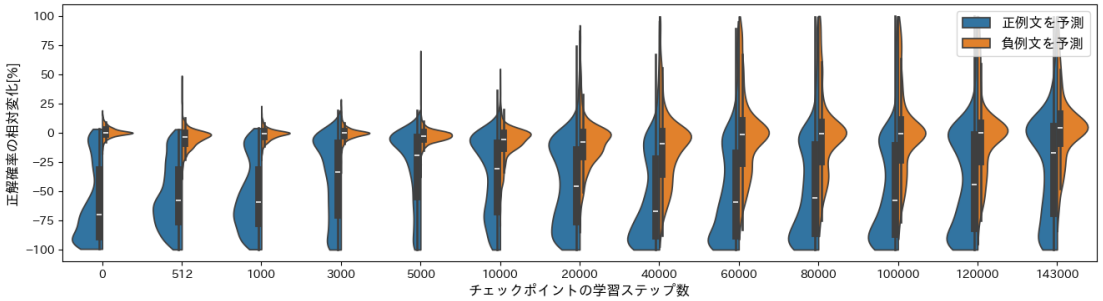

図 13: Pythia-410M について，各概念の寄与ニューロンの活性値を，適当な入力をした際の活性値に置き換えることで抑制した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

おいて知識ニューロンらしさの定義を満たすようなニューロンがあまり存在しないことが確認できる．

以上から，本研究で知識帰属法を各モデルに適用した結果，想定された “知識ニューロンの形成過程” を観察することはできなかった．この結果を踏まえ，我々は知識帰属法自体に関する調査が必要と判断し，いくつかの追加分析を行った．次章ではその分析結果と，その他に本章の実験に伴って判明したことについて報告する．

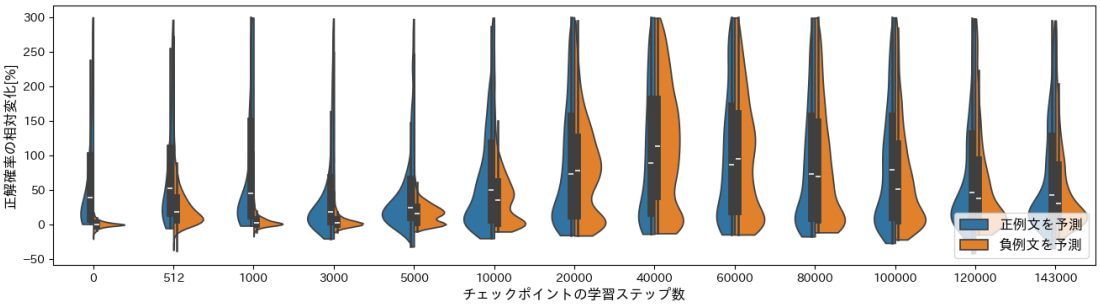

図 14: Pythia-410M について，各概念の寄与ニューロンの活性値を 2 倍に増幅した際の，文の続きに正例概念を予測する確率の相対変化率をモデルの学習ステップ数ごとに示したもの．

6　分析

6.1　学習ステップ数ごとに見つかる知識ニューロンの数

5.1 節の実験結果を取得するのと同時に，1 概念あたり平均で何個の知識ニューロンが見つかるかを，学習ステップ数ごとに調査した．その結果を図 15 に示す．横軸が学習ステップ数，縦軸が発見される 1 概念あたりの知識ニューロンの平均個数を表す．この図から，同じハイパーパラメータの実験設定であっても，学習ステップ数が 0，すなわちランダム初期化状態のモデルのみ，1 概念あたりに発見される知識ニューロンの数が突出して多いことが確認できる．この結果は，ランダム初期化の状態ではある概念の出力に大きな影響があるニューロンが数百個程度あるものの，一度学習が進むとそのようなニューロンが十数個程度に減少することを意味しているため，ある概念に反応するニューロンが学習によって一部のニューロンに特化する過程を示唆している．5.1 節の図 3，4，5 において，ランダム初期化状態のモデルの結果が他のチェックポイントと比較して特徴的であったのは，このような要素が影響していた可能性がある．

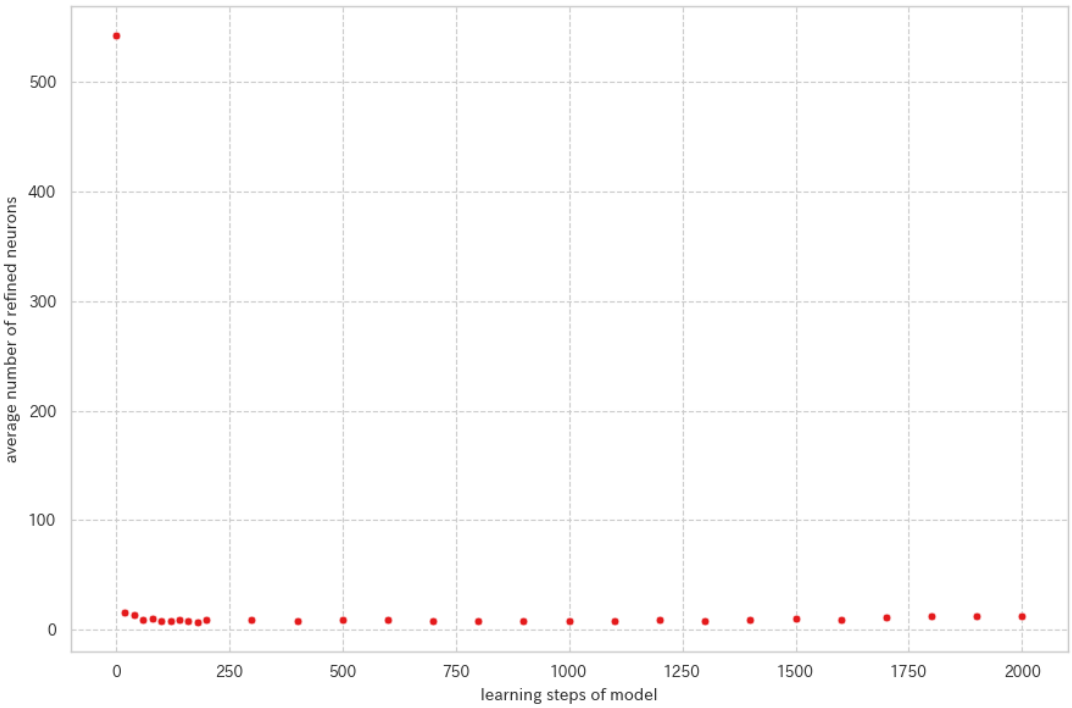

図 15: MultiBERTs の各チェックポイントについて，1 概念あたりに発見される知識ニューロンの個数の平均をプロットしたもの．

6.2 知識ニューロンが発見される場所と知識帰属法の関係

表 1 に示したように，本研究で使用した言語モデルは Transformer[3] を何層にも重ねて構成されている．そのため，あるニューロンは必ずそのいずれかの層内に存在しており，従って同じモデル内に存在するニューロンであっても入力層に近い層内に存在するニューロンと出力層に近いニューロンとが存在する．そこで，知識帰属法によって知識ニューロンとして発見されるニューロンがどの層に存在しているかを，MultiBERTs の再現元モデルである 12 層の BERT-base-uncased[7] モデルを対象に調査した[6]．この調査時に用いたデータセットは，MultiBERTs の

[6]MultiBERTs の学習ステップ数が最も大きいモデルではなくオリジナルの BERT を用いたのは，この分析においては学習過程を追跡するという要素がなく，そのような条件下ではより広く用いられているモデルを対象とした分析を行った方が得られる知見の価値が高いと判断したためである．

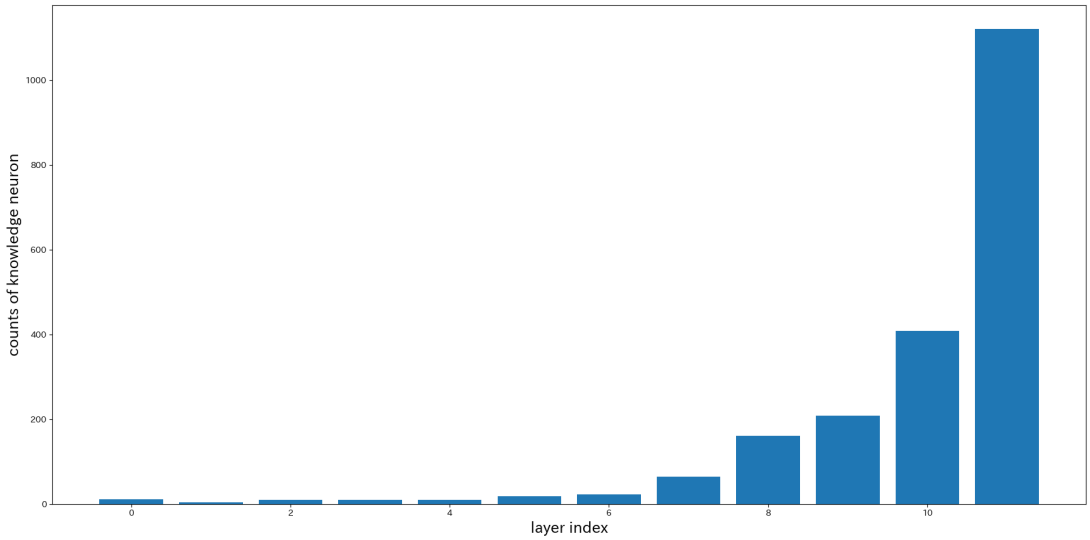

図 16: 知識帰属法によって発見される知識ニューロンが，どの層に存在しているかをカウントしたヒストグラム．

時と全く同じ GenericsKB から作成されたデータセットである．調査の結果を図 16 に示す．この図 16 は横軸が層のインデックス，すなわち左側の 0 に近いほど入力層に近い層で，右側の 11 に近いほど出力層に近い層を表している．縦軸は各層で発見された知識ニューロンのカウントを表す．この図 16 より，出力に近い層に存在するニューロンほど知識ニューロンとして発見される傾向があることが確認できる．この傾向が見られる原因として，出力に近いほどそれ以降のモデル内で行われる計算が少ないため，式 4 で積分によって測っている “ニューロンの活性値操作による出力確率への影響” が大きくなり，結果的に式 4 で計算される帰属値 Attr が大きくなってしまっているから，という仮説が考えられる．もしこの仮説が正しかった場合，知識帰属法はニューロンの存在する場所にバイアスを受ける手法ということになるため，本来の意味での知識ニューロンを探すための手法としては問題がある，ということになる．

以上を踏まえ，各層で発見される知識ニューロンの帰属値 Attr を集計し，層ごとに平均した結果を図 17 に示す．この図 17 より，単に出力に近い層で発見される知識ニューロンほど帰属値 Attr が高いわけではないことが確認できる．す

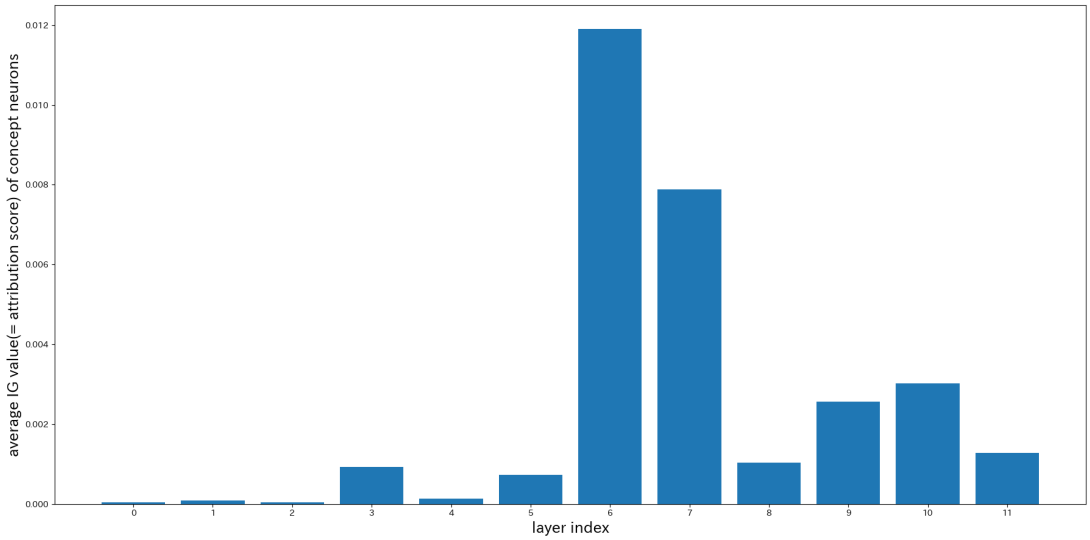

図 17: 式 4 によって計算される知識ニューロンの帰属値の，各層ごとの平均値を示したもの．

なわち，上述した仮説をサポートしない結果となっている．以上から，今回の分析ではなぜ出力に近い層ほど知識ニューロンが多く発見されるかの理由を明確にすることはできなかったものの，層の場所によって発見される知識ニューロンの数が異なることは興味深い結果であり，さらなる検証が必要である．

6.3　複数の概念をエンコードする知識ニューロン

本研究における概念の定義は 3.2 節で述べたように，あらゆる単語のことを指す．しかし表 1 内の “ニューロン総数” の列に示したように，モデル内に存在するニューロンの数には限りがあるため，全ての概念に知識ニューロンが存在すると仮定すると概念の方が総数が必然的に多くなり，あるニューロンがいくつもの概念の知識ニューロンになる場合が生じるはずである．そこで，本節でも 6.2 節と同様に BERT-base-uncased モデルに対して知識帰属法を適用し，ある知識ニューロンがいくつの概念の知識ニューロンとなっているかを調査した．調査の結果を

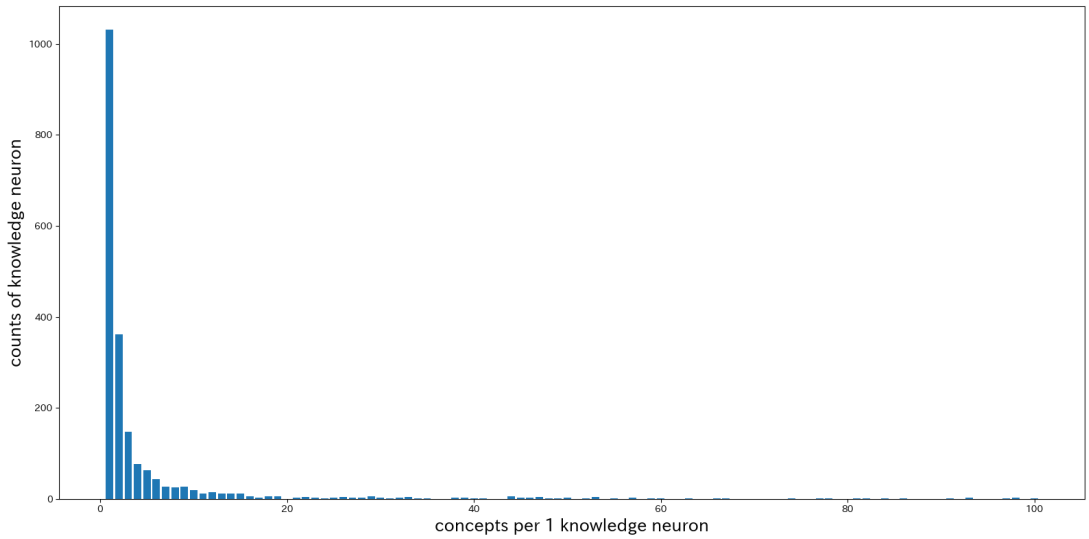

図 18: 1 つの知識ニューロンがエンコードしている概念の個数.

図 18 に示す[7]．図 18 から，確かに複数の概念をエンコードしている知識ニューロンが存在することが確認できる．

また追加の分析として，複数の概念をエンコードする知識ニューロンが具体的にどのような概念をエンコードしているかを目視によって確認した．その結果，一部の知識ニューロンは，エンコードしている複数概念に人間が解釈可能な共通点を持っていることが判明した．実際に共通点が存在すると判断できた事例の一部を表 3 に示す．この表 3 から，今回のデータセットで調査した概念の上位概念をエンコードする知識ニューロンの存在が示唆される．そのため，例えば “時”（=“time”）が入る穴埋め文を用いて探した知識ニューロンの集合に “11_2485” ニューロンが存在するかどうかを検証することで，意味としての “時” と表層形の “時” を言語モデルが内部できちんと関連付けられているかどうかを知ることができる可能性がある．

[7]実際には 101 概念以上をエンコードする知識ニューロンもごく僅かに存在したが，グラフの視認性を確保するために図 18 には 100 概念以下をエンコードする知識ニューロンの数を報告している．

表 3: 人間が解釈可能な共通点を持つ概念をエンコードする知識ニューロンの例. “知識ニューロン” 列はその知識ニューロンが存在するモデル内の場所を示しており，左側の数字が層の場所，右側の数字が層内でのインデックスを表す.

知識ニューロン	共通点	エンコードしている概念
11_2485	時	afternoon, evening, midday, tenth
10_36	人間	builder, collector, philosopher, traveler, wizard
9_3067	医学	surgery, therapy, insulin, medication, medicine, ...
10_2472	色	brown, white, yellow, good, slip
11_1062	体	eye, face, muscle, nerve, batter, porter
11_2480	ing 形	blowing, getting, judging, offspring, passing, ...

6.4 知識ニューロンの安定性

4 章の実験手順では，各チェックポイントにそれぞれ知識帰属法を適用するため，発見される知識ニューロンもチェックポイントごとに異なっている可能性が十分に考えられる．そこで本節では，そのようにして発見された知識ニューロンがどの程度チェックポイント間で重複しているかを調査した．具体的には，MultiBERTs モデルの 1 チェックポイントで発見される知識ニューロンが他のチェックポイントでも発見されるかを調査した．調査を行った中から代表して “japan” 概念について，学習ステップ数 2,000k のチェックポイントで発見される知識ニューロンが，他のチェックポイントでも発見されるか調査したものを図 19 に示す．この図は横軸がチェックポイントの学習ステップ数 ($[\times 10^3]$)，縦軸が学習ステップ数 2,000k のチェックポイントで発見された知識ニューロンのインデックスを表し，灰色の縦線はそのチェックポイントで “japan” の知識ニューロンを探したことを表している．この図 19 を分析すると，常に同じニューロンが “japan” の知識ニューロンになっているわけではないことが確認できる．例えば，“10_288” ニューロンは事前学習開始から 200k ステップのチェックポイントまでは “japan” の知識ニューロンになっていないが，300k ステップで一度 “japan” の知識ニューロンと判定された後，再度知識ニューロンでなくなっている．しかし，1,000k ステップからはま

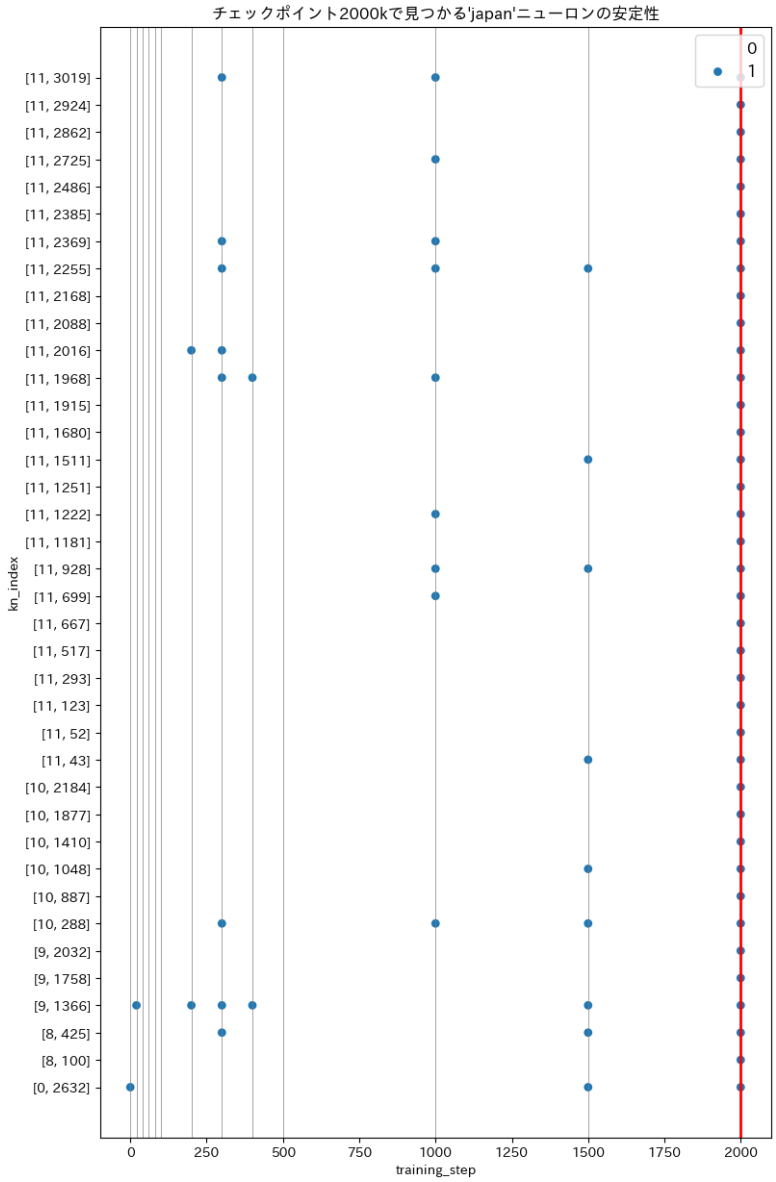

図 19: 学習ステップ数 2,000k のチェックポイントで発見される “japan” 概念の知識ニューロンと，その他のチェックポイントで見つかる “japan” 概念の知識ニューロンとの安定性.

た “japan” の知識ニューロンとなり，そのまま 2,000k ステップまで “japan” の知識ニューロンであり続けるという，複雑な経過を辿る結果を示している．一方でそのようなニューロンもあれば，“11_2924” ニューロンや “8_100” ニューロンは

2,000k ステップに学習が進むまで一度も “japan” の知識ニューロンと判定されていない．以上の結果から，知識ニューロンの形成過程において，必ずしも特定のニューロンが常に発達していく過程ではないことが確認された．

6.5　正例概念の出現頻度と学習過程

本研究では言語モデルが学習によって獲得する知識に着目しているが，直感的には言語モデルが学習中に何度も見るような概念の方が知識の獲得が早く行われると予想される．そこで本節では，概念の一般的な出現頻度とその知識獲得の早さに相関が見られるかを，pythia モデル（70M，160M，410M）を対象に調査し

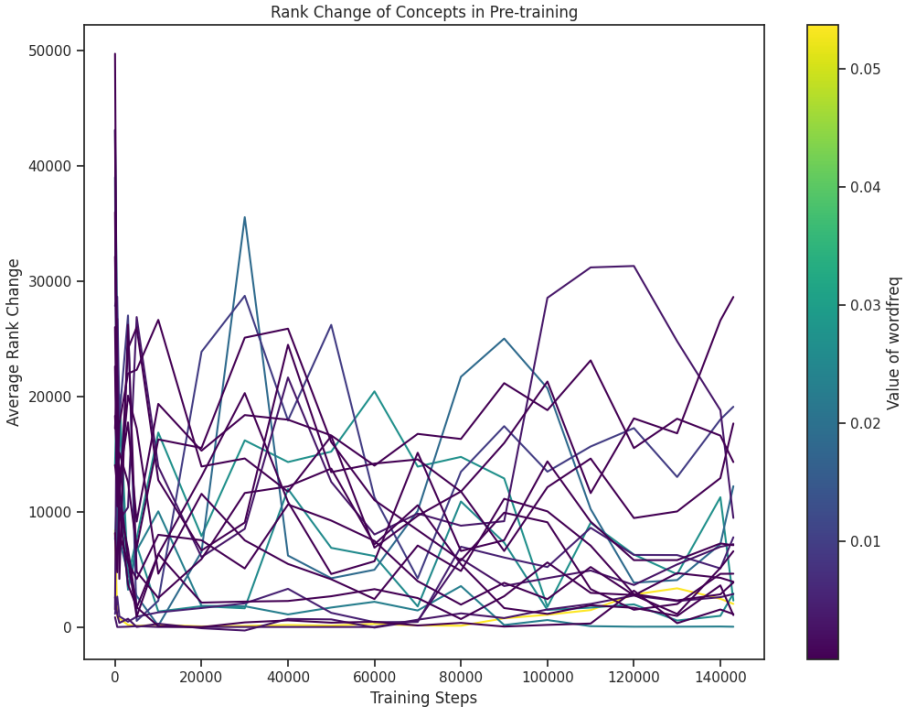

図 20: Pythia 70M について，概念の出現頻度と，寄与ニューロンの抑制操作による当該概念の学習過程に沿った出力順位変動．

た．調査方法の説明の前に，上述の直感に基づく次の仮説を立てる：

> **仮説**　出現頻度の高い概念ほど学習が早いならば，出現頻度の高い概念ほど，学習ステップ数が**より少ないチェックポイントの時点**で，その寄与ニューロンを抑制した時に，出力における当該概念の語彙上での出力順位が下がる傾向が見られる

この仮説のもと，以下の手順で調査を行う：

1. 1つのチェックポイントに知識帰属法を適用し，各概念の寄与ニューロンを見つける
2. そのチェックポイントに正例文を予測させ，その予測における正例概念の出

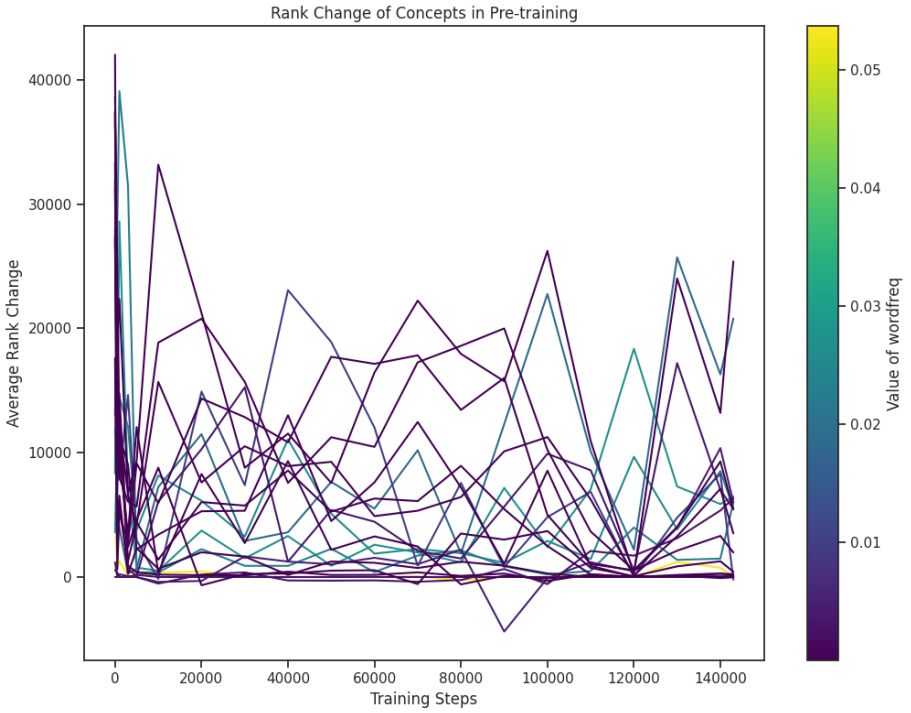

図 21: Pythia 160M について，概念の出現頻度と，寄与ニューロンの抑制操作による当該概念の学習過程に沿った出力順位変動.

力順位を記録する

3. そのチェックポイントの寄与ニューロンの活性値を 0 に抑制した上で再度正例文を予測させ，正例概念の出力順位を記録する

4. 手順 2. と 3. で記録した出力順位の差を計算し，折れ線グラフとしてプロットする

5. 手順 1. から 4. を，Pythia モデルの各チェックポイントに対して行う

但し，本調査において使用する概念は出現頻度に差があることが必要であるため，Natural Questions から作られたデータセットに含まれる概念のうち，一般的な出現頻度が最も高い 10 個と最も低い 10 個に限定した．なお，概念の一般的な出現頻度には wordfreq[26] による値を採用した．

上述した一連の調査手順によって得られた結果を，図 20，21，22 に示す．これらの図は，横軸が学習ステップ数，縦軸が抑制操作による出力順位の変動，右側のカラーバーが概念の出現頻度を表している．仮説が正しければ，

- 出現頻度の高い概念の結果である黄色から緑色にかけての折れ線グラフは，学習ステップ数が少ない時点から大きな順位変動を記録し続ける

- 逆に，頻度の低い概念の結果である青色から紫色にかけての折れ線グラフは，学習ステップ数が多くなってから大きな順位変動をするか，常に小さな順位変動しか記録しない

という傾向が読み取れるグラフになっているはずである．しかし実際は，図 20，21，22 のいずれを見ても，そのような傾向は確認することができない．従って，今回調査を行ったモデルの範囲では，概念の出現頻度と知識獲得の早さとの間に相関を見出すことはできなかった．

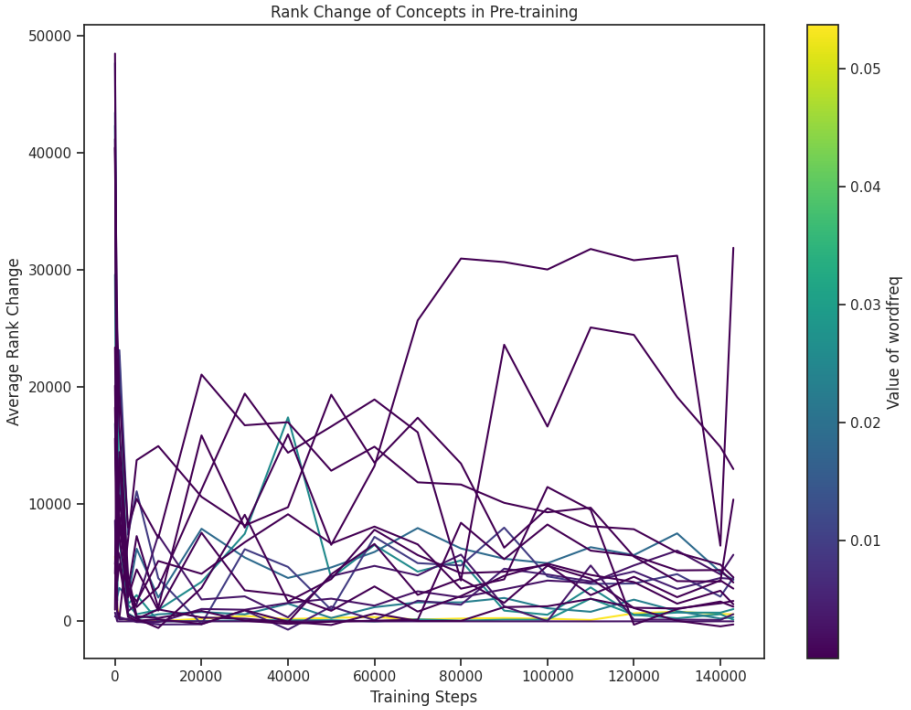

図 22: Pythia 410M について，概念の出現頻度と，寄与ニューロンの抑制操作による当該概念の学習過程に沿った出力順位変動.

7　議論

7.1　知識帰属法と知識ニューロン

本研究では，先行研究 [1] で提案された “知識帰属法” を用い，ある特定の概念の出力のみに影響を与えるとされる “知識ニューロン” が事前学習の過程でどのように形作られているのかを調査した．これは，直感的には知識の獲得が学習の進行に伴って行われるものであるという前提のもと，寄与ニューロンの “ある特定の概念の出力のみに影響を与える” という性質が次第に強まっていくことを確かめようとしたものである．しかし，実験結果は 5 章で示したように，“ある特定の概念の出力のみに影響を与える” という性質を持つニューロンは，マスク言語

モデルと生成型言語モデルの両方で学習ステップ数が少ない段階から存在することを示すものであった．本研究の結果と 3.3 節で説明した知識帰属法の手順を踏まえると，知識帰属法という手法はあくまで “ある特定の概念の出力のみに影響を与える” ニューロンを探し出すための手法に過ぎないものであると考えられる．

このように考えたとき，理想的な意味での「知識ニューロン」が言語モデル内に存在するかを調べるためには，本研究の実験設定で見直すべき点がいくつか存在する．まず，あるニューロンが複数の概念の知識をエンコードするようなケースは 6.3 節で示したように存在し得ると考えられる．そうであれば，“ある特定の概念の出力のみに影響を与える” という性質における” 特定の” という部分は，本実験の設定のように 1 つだけの概念を指すのではなく，複数の概念を指す意味で用いる方が自然である．次に，“知識の獲得が学習の進行に伴って行われる” という前提を踏まえると，知識ニューロンらしさには “学習の進行に伴ってその傾向が強まるもの” を定義する必要がある．すなわち，4.2 節で定義したような知識ニューロンらしさを追うことでは，本当には “知識を獲得していっているか” を測ることができない．仮に人間であれば，知識を獲得していく過程はその知識を問う問題の正解率の変化を学習段階に沿って観察することで測ることができると考えられ，そのような “知識を獲得しているかどうか” に基づく指標を用いた “知識ニューロンらしさ” を言語モデルにおいても定義する必要がある．これを具体的に定義する方法については，今後の課題である．

7.2 言語モデルと知識

本研究では，一貫して “言語モデルが持つ知識” に着目して実験や分析を行い，その中で “知識ニューロン” という用語を定義し用いた．しかし本研究で定義し使用した “知識ニューロン” は，3.3 節での発見手法を素朴に解釈すれば，厳密には “‘単語ニューロン” と呼んだ方が正確である．これは，“ある単語を出力する確率” という情報のみに基づいてニューロン探索手法が構築されていることに原因がある．例を挙げて説明すると，仮に言語モデルが「音」という概念[8]を全く理解

[8]7.2 節内で用いる「概念」とは本研究の中で定義した用語での使い方ではなく，一般的な意味である “あるものに対して抱く意味内容” という意味である．

していない，言わば「音」についての知識を持っていない状態であっても，モデルの事前学習は単語の出力確率が尤もらしくなるように行われるために “音” という単語の出力確率を増減するようなニューロンが存在し得り，本研究の実験設定ではそのような状態下のニューロンであっても知識ニューロンであると判定してしまう．この場合には “中国語の部屋” [27] の思考実験と同じ原理によって，実際には知識を保持していないニューロンが知識ニューロンであると捉えられてしまう．このような例が考えられるため，本研究では “穴埋め文が正しく埋められること” や “次単語を正しく予測できること” を「言語モデルが知識を持つこと」として扱うことにしている点は，本研究における重要な制限である．

また関連して，真に言語モデルに知識があるかどうかは定かではない．仮に言語モデルが知識を持つかどうかについて考えてみると，言語モデルは文字だけの情報をもとに学習を行っているため，例えば “色” の概念を本当に理解できているか，すなわち画像情報なしに “色” を理解できるか，といった問いに突き当たる．また別の観点からは，言語モデルは穴埋めや次単語予測といったタスクを確率的に尤もらしく行っているだけと捉えれば，言語モデルが学習しているのはその尤もらしい振る舞いであり，知識を獲得しているとは言えないと捉える立場も考えられる．いずれにしても，本研究が “言語モデルが言語を用いた事前学習タスクを通じて知識を獲得している” という立場と前提のもとに成り立っているものであることには注意が必要である．

8　おわりに

本研究では，事前学習済み言語モデルに存在するとされる知識ニューロンが，言語モデルの事前学習においてどのように形成されていくのか，その過程を調査した．それぞれマスク言語モデルと生成型言語モデルに対して実験を行った結果，“知識ニューロンらしさ” を持つニューロンが学習途中のチェックポイントからも発見され，既存手法である知識帰属法では理想的な意味での知識ニューロンを発見できていないことを示した．これらの結果を受け，知識帰属法自体に関する分析や，様々な観点からの知識ニューロンに関する分析を行い，“言語モデルに存

在する知識ニューロン” に関するいくつかの考察を行うことで，言語モデルと知識に関する議論を展開した．

今後の展望として，本研究では BERT[7] や GPT-3[22] をベースとしたモデルを研究対象としたが，その他の言語モデル，例えばより大規模なパラメータ数を持つ T5[28] といったモデルや，同じ GPT 系列モデルでもモデルサイズの小さい GPT-2[10] や，様々な自然言語処理タスクで高い性能を達成している GPT-4[11] といったモデルでどのような結果が得られるかについては，興味深い研究課題である．関連して，Pythia については 6.9B や 12B など，より大きなパラメータサイズのモデルが公開されており，それらのモデルでどのような知識ニューロンの形成過程が起こっているのか，本研究で実験を行ったモデルサイズによる結果との比較によって生じる違いを調査することが挙げられる．この調査によって，近年ニューラル言語モデルにおいて提唱されている法則である “スケーリング則”[29] のうち，モデルパラメータ数の増加による性能向上に関する新たな観点からの知見や，大規模言語モデルにおいて観察される現象であるとされる “創発的能力”[30] に関する知見を得ることができる可能性があり，関心が高まっている研究に関連した方向性の研究として期待される．

謝辞

本研究を進めるにあたり，多くの方々のご協力，ご助言をいただきました．お忙しいなか，研究活動だけでなく進路や学生生活に関してもご指導やご助言をいただいた主指導教員である乾 健太郎教授，鈴木 潤教授，坂口 慶祐准教授に心から感謝申し上げます．また，本論文の審査をお引き受けくださいました，本学の大林 武教授に深く感謝申し上げます．理化学研究所の Benjamin Heinzering さんには，日々の研究活動の際のご助言やご協力に加え，学生生活を送るうえでの多くのご助言をいただきました．心から感謝申し上げます．東北大学の穀田 一真さん，坂田 将樹さん，鴨田 豪さん，および Langsmith 株式会社の伊藤 拓海さんには，日々の研究活動の際のご助言やご協力をいただきましたことに，心から感謝申し上げます．最後に，研究会や日々の議論のなかで多くのご助言をいただきました Tohoku NLP Group の皆様，これまで学生生活や研究活動においてお世話になったすべての皆様に感謝申し上げます．

参考文献

[1] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

[2] 有山知希, Benjamin Heinzerling, 乾健太郎. Transformer モデルのニューロンには局所的に概念についての知識がエンコードされている. 言語処理学会第 28 回年次大会 発表論文集, pp. 599–603, 2022.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*, 2017.

[4] Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, Ian Tenney, and Ellie Pavlick. The multiberts: Bert reproductions for robustness analysis. *arXiv preprint arXiv:2106.16163*, 2021.

[5] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, KyleO ' Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff ほか. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

[6] Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*, 2017.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[8] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics.

[9] Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. Neurons in large language models: Dead, n-gram, positional, 2023.

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

[11] OpenAI. Gpt-4 technical report. *ArXiv*, Vol. abs/2303.08774, , 2023.

[12] Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. Language models can explain neurons in language models. `https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html`, 2023.

[13] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, Vol. 55, No. 12, pp. 1–38, 2023.

[14] 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, 2023.

[15] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. 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. 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[16] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022.

[17] 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.

[18] 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.

[19] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[20] 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 (EMNLP)*, pp. 5484–5495.

[21] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70, pp. 3319–3328, 2017.

[22] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry,

Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1877–1901. Curran Associates, Inc., 2020.

[23] Sumithra Bhakthavatsalam, Chloe Anastasiades, Peter Clark. Genericskb: A knowledge base of generic statements. *arXiv preprint arXiv:2005.00660*, 2020.

[24] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, Vol. 7, pp. 452–466, 2019.

[25] John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 119–126, 2020.

[26] Robyn Speer. rspeer/wordfreq: v3.0, September 2022.

[27] John R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, Vol. 3, No. 3, p. 417–424, 1980.

[28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67, 2020.

[29] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

[30] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022.

付録

A　活性値の抑制操作における入力

4.2 節で説明した寄与ニューロンの活性値抑制操作の中で，“活性値を，他の適当な入力をした際の活性値に置き換えることで抑制” という操作を説明した．ここでは，具体的に “適当な入力” に用いた文について補足する．

活性値置き換えによる抑制操作は MultiBERTs と Pythia の両モデルに対して行うことを考慮すると，適当な入力に望ましい性質は “両モデルに入力するデータセットと無関係であること” と考えられる．そのため，使用するデータセット内の全ての概念に含まれていない概念についての文が適当な入力に相応しいと判断した．これを踏まえた上で，一般的な単語である “key” という概念がデータセットに含まれていないことが確認できたため，それぞれ以下の文を “適当な入力” として使用した：

- MultiBERTs: “The [MASK], a small metal tool with notches and ridges, is used to unlock doors and access secured spaces.”
- Pythia: “what small, often metal object is used to unlock doors, start vehicles, or operate locks, and can come in different shapes and sizes to fit specific locks and mechanisms?”

発表文献一覧

受賞一覧

1. 人工知能学会 言語・音声理解と対話処理研究会 (SLUD) 第 99 回研究会 （第 14 回対話システムシンポジウム） 第 6 回対話システムライブコンペティション 最優秀賞

2. 言語処理学会 第 29 回年次大会 (NLP2023) 委員特別賞

3. 言語処理学会 第 28 回年次大会 (NLP2022) 富士通賞

4. 人工知能学会 言語・音声理解と対話処理研究会 (SLUD) 第 93 回研究会 （第 12 回対話システムシンポジウム） 第 4 回対話システムライブコンペティション 優秀賞

国内会議・研究会論文

1. 有山知希, 鈴木潤, 鈴木正敏, 田中涼太, 赤間怜奈, 西田京介. クイズコンペティションの結果分析から見た日本語質問応答の到達点と課題. 会誌「自然言語処理」, March 2024.

2. 中野雄斗*, 野末慎之介*, 穀田一真, 有山知希, 佐藤魁, 曾根周作, 亀井遼平, 謝素春, 成田風香, 守屋彰二, 赤間怜奈, 松林優一郎, 坂口慶祐. Hagi bot: LLM を用いた対話状態追跡と人間らしい振る舞いで自然な議論を行うマルチモーダル対話システム. 人工知能学会 言語・音声理解と対話処理研究会 (SLUD) 第 99 回研究会 第 14 回対話システムシンポジウム, November 2023. (*貢献は同じ)

3. 有山知希, Benjamin Heinzerling, 乾健太郎. 言語モデルの学習における知識ニューロンの形成過程について. 言語処理学会 第 29 回年次大会 (NLP2023), March 2023.

4. 有山知希, Benjamin Heinzerling, 乾健太郎. Transformer モデルのニューロンには局所的に概念についての知識がエンコードされている. 言語処理学会 第 28 回年次大会 (NLP2022), March 2022.

5. 長澤春希*, 工藤慧音*, 宮脇峻平, 有山知希, 成田風香, 岸波洋介, 佐藤志貴, 乾健太郎. aoba_v2 bot: 多様な応答生成モジュールを統合した雑談対話システム. 人工知能学会 言語・音声理解と対話処理研究会 (SLUD) 第 93 回研究会 第 12 回対話システムシンポジウム, November 2021. (*貢献は同じ)

6. 有山知希, Benjamin Heinzerling, 乾健太郎. BERT の世界知識はどこにある？学習済み言語モデルにおける知識の局所性を解明する. NLP 若手の会（YANS）第 16 回シンポジウム, August 2021.