

大規模言語モデルにおける日本語ゼロ照応解析能力の分析

野末慎之介¹ 石月由紀子^{1,2} 松林優一郎^{1,2} 坂口慶祐^{1,2}

¹ 東北大学 ² 理化学研究所

{nozue.shinnosuke.q5,yukiko.ishizuki.p7}@dc.tohoku.ac.jp

{y.m,keisuke.sakaguchi}@tohoku.ac.jp

概要

GPT-4 に代表される大規模言語モデルは言語処理の様々な下流タスクで好成績を収めることが明らかになっているが、その能力が真に言語の基礎的理解能力に由来するものであるかは明らかではない。本研究では、自然言語処理の基礎解析タスクの中でも困難なものの一つとされてきたゼロ照応解析タスクについて、大規模言語モデルの能力を評価する。実験の結果、GPT-4 はゼロ照応解析において十分な性能を発揮しないことが明らかになった。その原因を特定するための検証実験とエラー事例分析の結果から、ヲ格とニ格に関する格フレーム知識と、ニ格に関する選択選好知識の不足が原因であると示唆された。

1 はじめに

日本語では述語の項が頻繁に省略される。そのような省略された項 (ϕ) を同定するゼロ照応解析 (図 1) は、最先端のモデルにおいても F_1 値が 64% 程度であり [1], 解析が困難なタスクの一つである。一方で、GPT-4 [2] は、日本語理解能力を測るベンチマークの JGLUE において大規模言語モデルの中で最高性能を示し、特定のタスクで追加学習された BERT[3] などのモデルを一部タスクで上回った [4, 5]。日本語の文章にはゼロ照応表現が多く含まれるため、日本語理解に優れた GPT-4 はゼロ照応解析においても高い性能を発揮する可能性がある。

本研究では、日本語理解に最も優れた大規模言語モデルのゼロ照応解析性能を明らかにすることを目的として、同タスクの主要なベンチマークデータである NAIST Text Corpus (NTC) 1.5 [6, 7] における GPT-4 の性能を評価した。実験の結果、GPT-4 の性能は最先端モデルと比較して大きく劣ることが明らかになった。さらにいくつかの詳細な実験とエラー事例の分析により原因の考察を行ったところ、

$\phi = 私$

↓

私は花束を買ってから、(ϕ が) 会場に向かった。

図 1 ゼロ照応解析の例。省略された項は ϕ で表される。タスクでは、「向かう」のガ格に項があること、およびその照応先が「私」であることを推定している。

GPT-4 の解析性能は、ヲ格とニ格に関する格フレーム知識と、ニ格に関する選択選好知識の不足に起因することが示唆された。

2 日本語ゼロ照応解析

本研究のゼロ照応解析タスクは、主要なベンチマークコーパスである NTC 1.5 の仕様に従う。NTC では、ゼロ照応解析は述語の項を同定する述語項構造解析タスクの一部として内包されている。また、項の特定に関しては、ガ格、ヲ格、ニ格の必須格の項のみがアノテーションされているため [6, 7], 本研究でも、これらの 3 つの格について評価を行う。また、既存研究では一般に、述語と直接係り受けの関係にある項の解析を *dep* と呼び、ゼロ照応解析のうち、述語と項が同じ文内にある事例を *intra*, 述語と項が別の文にある事例を *inter*, 項が存在するものの、文書内に明示的に表出していない事例を *exophora* と呼び、区別している [1, 8, 9]。本研究での解析性能評価もこの区分に従う。

3 手法

3.1 GPT-4 によるゼロ照応解析

本研究では、項を GPT-4 に生成させることによりゼロ照応解析タスクを実現する。入力するプロンプトでタスクの指示と複数文で構成された文脈を与え、述語の位置と格を指定する。出力の形式は「〜」+ 格助詞 (が, を, にいずれか 1 つ) + 述語¹⁾

1) 「食べる」のガ格を問題とする場合は「〜が食べる」

とし、「〜」に入るフレーズを問う。出力されたフレーズの文脈における位置を特定し、その位置を解答として記録した。出力されたフレーズが文脈にない場合は *exophora* であるとみなし、そのフレーズが一人称を表すものであれば *author*、二人称を表すものであれば *reader*、この2つに当てはまらない場合は *general* としてその細分類を記録した。項が存在しないと判定した場合には「該当なし」と出力させた。また、共参照関係によって正解の項が文脈の複数の位置に含まれる場合はその全てを正解とした。

プロンプト プロンプトはタスク説明と文脈に分けられる。タスク説明では、タスクの指示、対象の述語の文脈における位置、1つの格を指定した。タスクの指示として受身形の述語は能動形として考えるよう指定した。これは、NTC 1.5 のアノテーションが能動形を基準に作られているためである。文脈は、述語の文とその前方3文を合わせた4文とした。文書において述語の前方が3文に満たない場合、後方から満たない行数文だけ抽出した。文書自体が4文に満たない場合は文書の全文を抽出した。文脈内では対象の述語を「**」で囲み、述語の位置を指定した。

few-shot 文脈内学習 5-shot の設定では *dep*, *intra*, *inter*, *exophora* の4つに、正解の項がない問題を加えた5つの例を見せることで文脈内学習を行った。5つの例はそれぞれ訓練セットから引用した。zero-shot, 5-shot のそれぞれのプロンプト例と出力例は付録の A に示す。

4 実験

ベースラインモデル 本研究では NTC 1.5 における最先端モデルである Konno ら [1] のモデルをベースラインとする。Konno らは bert-base-japanese を使用し、ゼロ照応解析を Cloze テスト形式の項の選択問題として定式化した。これは、系列の先頭に [CLS]、文の境界に [SEP] が挿入された複数の文から構成された系列 X について、その末尾に [MASK]+格助詞(が, を, にいずれか1つ) + 述語²⁾ で構成されるクエリフレーズを追加した系列 X' を入力とし、[MASK] に入るトークンを系列 X から選択させるというものである。トークンが系列 X 中に存在しない場合には、[CLS] が選択される。[CLS] が選択された場合は、4つのカテゴリ *author*, *reader*, *general*, *none* から1つを選択させる。 *author*, *reader*, *general*

2) 「食べる」のガ格を問題とする場合は「[MASK]が食べる」

表 1 NTC 1.5 の評価セットにおける F_1 値。太字は最良のスコアを示す。Konno らのスコアは論文より引用 [1]。GPT-4 は評価セットの1割 (2557 述語) で評価した結果であるため、厳密な比較はできない。

モデル	省略					
	ALL	<i>intra</i>	<i>inter</i>	<i>exophoric</i>	DEP	ALL
Konno+[1]	64.18	72.67	48.41	65.40	94.50	83.65
zero-shot	30.54	32.64	29.38	13.79	54.14	46.42
5-shot	32.78	35.19	30.60	20.14	55.65	48.19

表 2 5-shot における格ごとの適合率, 再現率, F_1 値。

格	適合率	再現率	F_1 値
ガ格	65.12	63.61	64.36
ヲ格	35.46	83.09	50.32
ニ格	8.39	74.55	15.05

は上述の通り、*exophora* の細分類であり、*none* は該当する項が存在しないことを表す。

実験設定 GPT-4 の評価は zero-shot と 5-shot で行った。使用した GPT-4 のバージョンとパラメータは付録の B に示す。

データセット分割 Konno ら [1] に倣い、データセットは Taira らの分割法 [10] に従って訓練・開発・評価セットの3つに分割した。GPT-4 のプロンプトエンジニアリングには訓練・開発セットを用い、評価には評価セットを用いた。

4.1 結果

結果を表 1 に示す。ただし、GPT-4 の値は評価セットの1割の事例数で評価した段階で顕著にベースラインを下回っていたため、この時点で評価を終了した。したがって、表の数字は評価セット内のランダムな1割の事例で算出したものである。zero-shot, 5-shot のいずれも全項目でベースラインを大幅に下回った。その一方で、5-shot では、zero-shot と比較して全項目でスコアの改善が見られたことから、文脈内学習の効果はあるといえる。

GPT-4 の解析能力の低さををより詳細に分析するため、5-shot モデルの4カテゴリ全体における格ごとのスコアを表 2 に示した。特にヲ格とニ格については、再現率と比較して適合率が著しく低いことが明らかになった。この理由をさらに詳しく分析するため、GPT-4 のエラーを、1. 正解の項があるが誤った項を出力、2. 正解の項があるが「該当なし」と出力、3. 正解の項がないが項を出力の3つに分類した (表 3)。分類の結果から、3のパターンが最多のエラーであることが明らかになり、GPT-4 は、本

表3 GPT-4 が誤った事例の分類.

エラーの種類		zero-shot	5-shot
正解の項あり	1. 誤った項を出力	1,109	1,075
	2. 「該当なし」と出力	147	80
正解の項なし	3. 項を出力	3,652	3,583

表4 GPT-4 が誤った事例. 太字は対象の述語を示し, 中央の列は判別対象の格を示す.

文脈	格	出力
チェチェン共和国に隣接するグルジアで...	ヲ格	チェチェン共和国を隣接する
... 官房長官は「聞いていて涙が出た」...	ヲ格	涙を出す
... 白書は... と述べている。	ニ格	「白書に述べる」

来存在しない格助詞を必要以上に回答するエラーを多数引き起こしていることが分かった. 次節ではこの要因を特定するために更に詳細な分析を行う.

5 エラー事例分析

上述した3のパターンの出力例を表4に示す. まず, 1行目と2行目の事例では, 述語がどの文脈でもとり得ない格を判別できていない. 1行目の事例の「隣接する」はヲ格をとらない. また, 2行目の例では, 「出る」は「内側から外へ移動する.³⁾という語義で, 「店を出る」のようにヲ格をとる場合もあるが, この文脈の「物が産出される.³⁾という語義で使われる場合にはヲ格をとらない. このように, 述語がどの格をとり得るかは, その述語の性質と語義によって決まる. 述語がどの格をとる可能性があるか記録したものを格フレームといい, 本研究ではこの知識を**格フレーム知識**と呼ぶ. これらの事例からGPT-4の格フレーム知識の不足が疑われる.

次に, 3行目の事例では, 述語はとり得るが, 対象の文脈ではとらない格を判別できていない. 通常, 「述べる」のニ格には相手を表す言葉が入り, 場を表す言葉はデ格の項になる. しかし, この事例ではニ格に「述べる」場を表す言葉として「白書」が入っている. このことから, GPT-4はニ格とデ格でとる言葉の属性を混同していると推測される. この述語が格によってどのような属性の言葉をとるかについての選好性を**選択選好知識**といい, 本研究ではこの知識を**選択選好知識**と呼ぶ. このような事例からGPT-4の**選択選好知識**は不足していることが疑われる.

以上から6節では, 4.1節の実験におけるヲ格, ニ

3) スーパー大辞林を参考に作成.

表5 GPT-4による述語の格フレーム理解度の検証結果.

格助詞	適合率	再現率	F_1 値
ガ格	100.00	97.00	98.48
ヲ格	84.72	98.39	91.04
ニ格	74.63	76.92	75.76
ALL	88.14	91.63	89.85

格の適合率が低いことが, 格フレーム知識と**選択選好知識**の不足に起因するかを検証する.

6 知識の検証

本実験は2段階で構成される. まず, 入力プロンプトで述語, 語義, 格をそれぞれ1つずつ指定し, 指定された語義に対して述語がその格をとるかどうかを「はい」か「いいえ」を出力させた. この出力を正解の格フレームで正誤判定し, 格フレーム知識の理解度を測る. 次に最初の問いに対する回答が「はい」である場合に, 述語が対象の格をとる例を3文出力させた. 出力のうち, 例として自然な文章の割合によって, **選択選好知識**の理解度を測る. プロンプトと出力の例は付録のDに示す.

6.1 実験設定

評価方法 格フレーム知識に対する評価には F_1 値を用いた. 続いて, **選択選好知識**に対する評価は, 人手評価により正解率を求めた. ただし, 正解率の算出には最初の問いで正解ラベルが「はい」である事例(以下, 述語が対象の格をとる事例)のみを用いたが, エラー事例の分析には, 正解ラベルが「いいえ」である事例(以下, 述語が対象の格をとらない事例)を含めた.

データセット NTC 1.5の評価セットからランダムに抽出した100個の異なる述語を問題として扱った. 正解となる述語の格フレームは, 京都大学格フレーム[11]に基づき, 事前に著者ら3名によって定められた. 京都大学格フレームは語義ごとに格フレームが分類されているため, 述語の語義を特定することで適切な格フレームが定まる. そこで語義は述語の文脈とスーパー大辞林を参照して特定した. データセットの作成手順は付録のCに示す.

6.2 格フレーム知識の検証結果

結果を表5に示す. ガ格とヲ格の F_1 値は90%を超えており高いものといえるが, ニ格は他の格に比べて15%以上低く, 相対的にニ格に関する格フレーム知識が不足しているといえる.

6.3 選択選好知識の検証結果

正解率はガ格が 90.72%, ヲ格が 92.35%, ニ格が 80.67%, 全体で 88.78%であった。ガ格, ヲ格に関する選好選択知識の正解率が 90%以上である一方で, ニ格は他の格に比べて 10%以上正解率が低く, ニ格に関する選好選択知識の理解度は相対的に低い。

6.4 検証におけるエラー事例分析

まず述語が対象の格をとる事例における主な誤りは, 他の語義であればその格でとるが, 指定した語義ではその格でとらないようなフレーズを答えた事例であった。例えば, 「終わりにする。」³⁾という語義の「閉じる」のヲ格では「生涯を閉じる」のように, 終わりとなる対象を表す項をとる。しかし, GPT-4 は「ドアを閉じる」のような「開いていたものをふさがった状態にする。」³⁾という語義の例を生成しており, 一部の述語では語義によって項の選択選好性を識別できていない。

次に述語が対象の格をとらない事例における主な誤りは, 述語が他の格のフレーズとしてとるものを, 述語がとらない格の項として答えた事例であった。例えば「全壊する」という述語は「家」という言葉に対してガ格をとり, いずれのフレーズに対してもヲ格をとらないが, 「家を全壊する」というような不自然な日本語が生成された。このことから, 一部の述語では, その述語がとりうる格, およびその格に対する適切な選択選好性を理解できていないといえる。特にヲ格においては, この事例がほとんどであった。また, ニ格では係り受け関係を理解していないような事例も見られた。例えば「非難する」のニ格の例として「彼女は社長に非難するためのレターを送った。」という文が生成された。この「社長に」は「非難する」ではなく, 「送った」のニ格であり, 係り受け関係を誤読していることがわかる。

7 考察

6 節の結果から 4.1 節の結果の原因を検討する。ただし, GPT-4 のガ格の格フレームと選択選好に対する理解度は高く, 今回の仮説以外の原因がガ格に対するゼロ照応解析性能に影響することが示唆されたため, 本節ではヲ格とニ格について議論する。

まず表 5 から, ヲ格とニ格の適合率が低さが明らかになった。これは GPT-4 がヲ格やニ格をとると認識している述語のうち, それぞれ約 15%と約 25%の

述語が誤っていることを示す。6.4 節から, GPT-4 がヲ格やニ格に関する格フレームを誤る原因として, 他の格でとるフレーズをヲ格やニ格でとるフレーズだと誤認していることがいえる。また, ニ格においては, ニ格に関連する係り受け関係を誤読してしまっていることから, 格フレームに関連する他の基礎的な知識も不足していることが示唆された。以上のことが原因となり表 4 の 1, 2 行目のようなエラーが起きていると考えられる。

次に 6.3 節から, ニ格では約 20%の述語で項の選好性に対する認識が誤っており, ニ格をとると正しく推測した場合でも文脈から誤った項を選択する可能性が他の格に比べて高いといえる。6.4 節より, その主な原因は語義ごとに異なる項の選好性を理解できていないことであるといえ, このような選択選好知識の不足により表 4 の 2 行目のようなエラーが起きると考えられる。

以上の議論から, GPT-4 にはヲ格とニ格に関する格フレーム知識と, ニ格に関する選択選好知識が不足していることがヲ格とニ格におけるゼロ照応解析性能を下げていると考えられる。

最後に本実験とは直接関係しないが, データセット作成時に, NTC 1.5 上でニ格がアノテーションされていない事例を発見した。これは NTC 1.5 のポリシーによるものだと考えられる。今回抽出した述語のうち, ニ格がアノテーションされていない述語は 89 個であるが, そのうち 16 個はこのような事例であった。そのため 4.1 節におけるエラーのうち, 一部は違うポリシーであれば正解になる可能性がある。しかしながら, そのような事例を除いたとしてもエラー数は依然多いため, エラーの根本的な原因を探索する試みには大きな意義があるといえる。

8 おわりに

本研究では GPT-4 を自然言語処理の基礎解析タスクの 1 つであるゼロ照応解析で評価し, 最先端モデルに対して性能が著しく劣ることを明らかにした。さらにその原因を解明すべく, エラー事例分析により格フレーム知識と選択選好知識の不足が原因であるという仮説を立て, 実験を行った。実験の結果と分析から, GPT-4 のヲ格とニ格に関する格フレーム知識と, ニ格に関する選択選好知識の不足により, ゼロ照応解析の性能を下げていると示唆された。

謝辞

本研究の有意義なコメントをいただいた Tohoku NLP Group の守屋彰二氏，塩野大輝氏，矢野一樹氏に感謝いたします。本研究は JSPS 科研費 21K21343 の助成を受けたものです。

参考文献

- [1] Ryuto Konno, Shun Kiyono, Yuichiroh Matsubayashi, Hiroki Ouchi, and Kentaro Inui. Pseudo zero pronoun resolution improves zero anaphora resolution. 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. 3790–3806, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [2] OpenAI. GPT-4 technical report. March 2023.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
- [4] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In **Proceedings of the Thirteenth Language Resources and Evaluation Conference**, pp. 2957–2966, Marseille, France, June 2022. European Language Resources Association.
- [5] 栗原健太郎, 河原大輔, 柴田知秀. Jglue: 日本語言語理解ベンチマーク. 言語処理学会第 28 回年次大会, 2022.
- [6] Ryu Iida, Mamoru Komachi, Kentaro Inui, and Yuji Matsumoto. Annotating a Japanese text corpus with Predicate-Argument and coreference relations. In **Proceedings of the Linguistic Annotation Workshop**, pp. 132–139, Prague, Czech Republic, June 2007. Association for Computational Linguistics.
- [7] Ryu Iida, Mamoru Komachi, Naoya Inoue, Kentaro Inui, and Yuji Matsumoto. **NAIST Text Corpus: Annotating Predicate-Argument and Coreference Relations in Japanese**, pp. 1177–1196. Springer Netherlands, Dordrecht, 2017.
- [8] Tomohide Shibata and Sadao Kurohashi. Entity-centric joint modeling of Japanese coreference resolution and predicate argument structure analysis. In Iryna Gurevych and Yusuke Miyao, editors, **Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 579–589, Melbourne, Australia, July 2018. Association for Computational Linguistics.
- [9] Masato Umakoshi, Yugo Murawaki, and Sadao Kurohashi. Japanese zero anaphora resolution can benefit from parallel texts through neural transfer learning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, **Findings of the Association for Computational Linguistics: EMNLP 2021**, pp. 1920–1934, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [10] Hiroto Taira, Sanae Fujita, and Masaaki Nagata. A Japanese predicate argument structure analysis using decision lists. In Mirella Lapata and Hwee Tou Ng, editors, **Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing**, pp. 523–532, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.
- [11] 大輔河原, 禎夫黒橋. 高性能計算環境を用いた web からの大規模格フレーム構築. 情報処理学会研究報告 = IPSJ SIG technical reports, Vol. 2006, No. 1, pp. 67–73, 01 2006.

A ゼロ照応解析実験のプロンプト

5-shot, zero-shot それぞれのプロンプト例とその出力を表6と表8に示す。

表6 zero-shot におけるプロンプトの例。本文部分は9510ED-0016-951012044.ntcより抜粋。

プロンプト: 以下の文章内で、「**」で囲まれた「出し」という述語は、誰を、もしくは何を、出すか分かるでしょうか。もし、「～を出す」に当てはまるフレーズがあれば、そのフレーズを「～を出す」という形式で教えてください。述語が受け身形の場合は、原型の述語に対する「～を出す」の形で考えてください。「～を出す」という表現が不自然だったり、具体的なフレーズがこの本文からでは推測できない場合は「該当なし」と教えてください。【本文】事件報告を受けた直後の大蔵省の事情聴取も、大和銀施設で食事、ビール付きで行われたという。大蔵省は大和銀に対して、近く業務改善命令を出すという。業務改善命令は銀行法に基づく行政処分であり、都市銀行に発令するのは初めてのこととなる。一連の不正事件に照らせば当然であるが、大蔵省についても、国民は「業務改善命令」を**出し**たいところだ。答えは

出力: 「業務改善命令を出す」

B GPT-4のバージョンとパラメータ

本研究で使用したモデルのバージョンはgpt-4-0613、各パラメータの値はtemperatureが1.0、Top Pが0.8、Max lengthが100であった。

C 格フレーム知識問題の作成手順

以下の(1)から(5)までの操作を100回(100述語分)完了するまで繰り返す。(1)述語をNTC 1.5のtestからランダムに抽出。(2)述語が「AはBだ」のBであるような場合と、抽出した述語の原型の表層形と同じ述語がすでに問題として作成されている場合は操作を終了して1に戻る。(3)述語をスーパー大辞林で検索し、ガ格・ヲ格・ニ格のヒントが含まれていないような語釈を抽出し、問題の手がかりとなる部分は削除・変更する⁴⁾。この際、スーパー大辞林の検索でヒットしない場合操作を終了して1に戻る。また、格フレームの手がかりとなる部分を削除する操作を加えると語釈が歪んでしまう場合は操作を終了して(1)に戻る。(4)格フレームで述語を検索し、3で抽出した語釈の場合の格フレームを特定し、正解を作成。この際、語が格フレームに収録されていない場合、もしくは語釈に適した格フレームがない場合は操作を終了して(1)に戻る。(5)ガ格・ヲ格・ニ格、それぞれを聞くプロンプトを作成。この手順は著者ら3名で行った。まず、1名が一連の操作を行った後、(5)で作成した正解が妥当なものであるか2名が確認して適宜修正した。

D 知識検証のプロンプト

GPT-4の格フレーム知識と選択選好知識を検証する際に使用したプロンプトとその出力例を表7に示す。

4) 例えば、「少ない」はスーパー大辞林において「数や量が小さい。」という語釈文であるが、「数や量が少ない」と置き換えても意味が変化しないため、「少ない」がガ格をとると推測できてしまう。このような手がかりはGPT-4が本来持っている知識を引き出す問題としてノイズになるため削除した。

表7 知識の検証実験に用いたプロンプト。

プロンプト: 「笑う」という述語が「に」という格助詞をとるかどうかを「はい」「いいえ」で教えてください。「はい」の場合は具体例を3つ教えてください。ただし、「笑う」は「あざける。嘲笑する。」という意味の場合を考えてください。また、格助詞をとるとは、格助詞+述語という表現ができることを表し、具体的には「～に笑う」という表現が日本語としてあり得るものかを聞きたいです。しかし、「先に」や「帰りに」といった時間を表す表現や、「～ように」や「遠巻きに」といった様態を表す表現は「に」をとるとみなしません。

出力: はい

1. 彼は彼女の失敗に笑った。
2. 私は彼の冗談に笑った
3. 私たちはその皮肉なコメントに笑った。

表8 5-shot におけるプロンプトの例。

プロンプト: 以下の文章内で、「**」で囲まれた「見て」という述語は、誰を、もしくは何を、見るか分かるでしょうか。もし、「～を見る」に当てはまるフレーズがあれば、そのフレーズを「～を見る」という形式で教えてください。述語が受け身形の場合は、原型の述語に対する「～を見る」の形で考えてください。「～を見る」という表現が不自然だったり、具体的なフレーズがこの本文からでは推測できない場合は「該当なし」と教えてください。【本文】

この数字を**見て**ほしい。答えは数字を見る

以下の文章内で、「**」で囲まれた「知ら」という述語は、誰を、もしくは何を、知るか分かるでしょうか。もし、「～を知る」に当てはまるフレーズがあれば、そのフレーズを「～を知る」という形式で教えてください。述語が受け身形の場合は、原型の述語に対する「～を知る」の形で考えてください。「～を知る」という表現が不自然だったり、具体的なフレーズがこの本文からでは推測できない場合は「該当なし」と教えてください。【本文】

日本の政府開発援助の総額が、世界最大級だとよく**知ら**れている。答えは総額を知る

以下の文章内で、「**」で囲まれた「言えば」という述語は、誰を、もしくは何を、言うか分かるでしょうか。もし、「～を言う」に当てはまるフレーズがあれば、そのフレーズを「～を言う」という形式で教えてください。述語が受け身形の場合は、原型の述語に対する「～を言う」の形で考えてください。「～を言う」という表現が不自然だったり、具体的なフレーズがこの本文からでは推測できない場合は「該当なし」と教えてください。【本文】

「言うは銀、黙すは金」ということわざが英語にある。日本語流に**言えば**「言わぬは言うにまざる」。答えはことわざを言う

以下の文章内で、「**」で囲まれた「訂正」という述語は、誰を、もしくは何を、訂正するか分かるでしょうか。もし、「～を訂正する」に当てはまるフレーズがあれば、そのフレーズを「～を訂正する」という形式で教えてください。述語が受け身形の場合は、原型の述語に対する「～を訂正する」の形で考えてください。「～を訂正する」という表現が不自然だったり、具体的なフレーズがこの本文からでは推測できない場合は「該当なし」と教えてください。【本文】

お呼びして**訂正**します。答えは誤りを訂正する

以下の文章内で、「**」で囲まれた「なる」という述語は、誰を、もしくは何を、なるか分かるでしょうか。もし、「～をなる」に当てはまるフレーズがあれば、そのフレーズを「～をなる」という形式で教えてください。述語が受け身形の場合は、原型の述語に対する「～をなる」の形で考えてください。「～をなる」という表現が不自然だったり、具体的なフレーズがこの本文からでは推測できない場合は「該当なし」と教えてください。【本文】

もう二十日近くに**なる**。

以下の文章内で、「**」で囲まれた「出し」という述語は、誰を、もしくは何を、出すか分かるでしょうか。もし、「～を出す」に当てはまるフレーズがあれば、そのフレーズを「～を出す」という形式で教えてください。述語が受け身形の場合は、原型の述語に対する「～を出す」の形で考えてください。「～を出す」という表現が不自然だったり、具体的なフレーズがこの本文からでは推測できない場合は「該当なし」と教えてください。【本文】

事件報告を受けた直後の大蔵省の事情聴取も、大和銀施設で食事、ビール付きで行われたという。大蔵省は大和銀に対して、近く業務改善命令を出すという。業務改善命令は銀行法に基づく行政処分であり、都市銀行に発令するのは初めてのこととなる。一連の不正事件に照らせば当然であるが、大蔵省についても、国民は「業務改善命令」を**出し**たいところだ。答えは

出力: 業務改善命令を出す