pfgen-bench: 日本語事前学習モデルのための文章生成性能評価ベンチマーク

今城 健太郎 [1,2,†,*]　平野 正徳 [1,2,†]　鈴木 脩司 [1,2]　三上 裕明 [1,2]
[1] 株式会社 Preferred Networks　[2] 株式会社 Preferred Elements
imos@preferred.jp　research@mhirano.jp
{ssuzuki, mhiroaki}@preferred.jp

概要

本研究では、日本語事前学習モデルの文章生成性能を評価するためのベンチマークである pfgen-bench を提案する。従来の大規模言語モデル (LLM) を用いた日本語生成性能評価は、回答の正確性などに重きが置かれており、LLM-as-a-judge のような生成の良さを見るようなベンチマークであっても、英語での回答を高く評価してしまい、日本語としての流暢さが十分に評価されないという問題点がある。そこで、本手法では、Fluency(流暢さ)、Truthfulness(真実性)、Helpfulness(有用性) の 3 つの評価軸からなるベンチマークを提案する。まず、日本の小中高の学習指導要領を参考に、13 科目 50 問からなる、日本語圏特有の常識問題集を作成した。さらに、複数の LLM とルールベースのフィルタリング手法を用いて、高品質な参照回答群を構築した。その上で、さまざまなモデルの回答と参照回答群の近さをはかる 3 つの評価軸を設計し、生成結果の評価が可能なベンチマークを構築した。このベンチマークを使用した評価結果に基づくと、事前学習モデル間の性能差を明確に示し、LLM による従来の評価とも一致する点が確認された。構築したベンチマークは公開し、一般に使用可能である。

キーワード: 大規模言語モデル、日本語、ベンチマーク、評価

1　はじめに

近年、事前学習された大規模言語モデル (LLM) は、翻訳や文章生成、質問応答などの自然言語処理タスクにおいて高い性能を示している。特に、Llama[1, 2] や Mixtral[3] などのモデルは、英語圏を中心に広く活用され、その汎用的な能力が評価されている。一方で、これらのモデルは主に英語データを基に訓練されており、日本語のような非英語圏の言語に対して同様の性能を発揮するかどうかは十分に検証されていない。

従来の日本語生成能力の評価手法にはいくつかの問題点が存在する。多くのベンチマークが選択肢形式の問題を採用しているが、この形式ではモデルの日本語生成能力が不十分であっても、日本語の読解力や英語の理解を駆使して正答を得ることができるため、生成能力を適切に評価することが難しい。また、Japanese MT-bench[1)] のような日本語の生成能力を計測するベンチマークでも、英単語や外来語を多用した日本語として不自然な回答に対しても減点があまり行われず、日本語としての流暢さや文脈適合性を十分に評価できないという課題がある。また、生成能力の評価にあたっては、非常に性能の高い言語モデルを評価者として用いる必要もあり、計算コストの観点でも課題が存在する。

加えて、モデルの性能は与えられたプロンプト (指示文) に大きく依存する。プロンプトの設計がわずかに異なるだけで、同じモデルが大きく異なる応答を生成することがあり、プロンプト依存性が高い評価では、モデルの潜在能力を正確に評価することが困難となる。そこで本研究では、指示文に用いず、質問と回答を列挙することで、モデルの応答形式を間接的に示すことで、指示学習が行われていない事前学習モデルに対して公平な評価を行う手法を提案する。

本研究の目的は、日本語事前学習モデルの生成能力を小規模なモデルから大規模なモデルまで様々な規模のモデルをより高い分解能で評価するための

* Corresponding Author: imos@preferred.jp
† Equal contribution

1) https://github.com/Stability-AI/FastChat/tree/jp-stable

ベンチマーク手法を提案することである。本ベンチマークでは、Fluency(流暢さ)、Truthfulness(真実性)、Helpfulness(有用性) の 3 つの評価軸を採用し、日本語モデルがどの程度正確で流暢な応答を生成できるかを評価する。これにより、日本語生成モデルが日本語の文脈に適応し、適切な回答を生成できるかを客観的に評価することが可能となる。

また、本研究で提案する評価手法は日本語モデルに特化しているが、他の言語や分野にも応用可能である。英語や他言語の生成能力、あるいはプログラミングコード生成能力を評価するためのベンチマークとしての応用も考えられる。

本研究で構築した pfgen-bench は https://github.com/pfnet-research/pfgen-bench で公開している。

2 関連研究

大規模言語モデル (LLM) は、近年、著しい発展を遂げている。特に、ChatGPT[4] や GPT-4[5]、GPT-4o をはじめとした最新の言語モデルは、性能向上と汎化が著しい。その基本技術は Transformer[6] から始まっており、BERT や [7] や、GPT シリーズ [8, 9, 10] などが続いた。ほかにも、Bard[11] や LLaMA[1, 2]、Dolly[12]、BLOOM[13]、Vicuna[14]、PaLM[15, 16] などのモデルが提案されている。

しかしながら、これらの大規模言語モデルは、様々なタスクでどの程度の性能を発揮するかは未知数であり、それらを評価する取り組みが進められている。たとえば、Language Model Evaluation Harness (lm_eval) [17] と呼ばれる、LLM 用の様々なタスクによるベンチマーク計測プラットフォームが提案されている。さらに、知識理解を包括的に問う MMLU[18] や推論能力を見る MMLU-Pro[19] 及び GPQA[20]、数学の性能を計るベンチマーク [21, 22], 多言語での思考能力を見る MGSM[23]、プログラミングの能力を見る Human Eval [24] や MBPP[25] など、様々なベンチマークが提案されている。加えて、GPT-4[5] においても、様々なタスクにおける性能を評価している研究もある。たとえば、会計士試験の達成度 [26] や医学分野における応用 [27]、法律分野への応用 [28, 29] を検証する研究などが存在する。

こうした、ドメインやタスク特化のベンチマーク以外にも、生成の良さ、つまり、AI アシスタントとしての有用性や質問応答の適切さにフォーカスを当てたベンチマークも存在する。MT-bench[30] では、強力な性能を持つ LLM を評価者 (LLM-as-a-judge) として用いることによって、複数ターンにおける会話応答の適切性を評価するベンチマークを提案しており、この LLM-as-a-judge は広く使用されている。また、Chatbot Arena [31] では、実際に人間が出力の良さを二対比較で評価し続けることで、LLM の性能を比較するプラットフォームを提案している。

これらのベンチマークの重要性を鑑み、本研究においても、新しい、生成の良さを見るベンチマークを提案する。

3 提案ベンチマーク: pfgen-bench

本章では、pfgen-bench を提案する。提案ベンチマークは、事前学習モデルにおける、日本語の生成能力および日本語圏特有の常識の習熟度を計測することを目的としている。一方で、従来の LLM-as-a-judge [30] のような特定の LLM に依存したり、プロンプトによる性能評価の違いをなくすことも併せて目的としている。

提案ベンチマークの構築にあたっては、大きく分けて以下の 3 つのステップが存在する。

- 問題・回答例構築
- 参照回答群構築
- モデルの評価値計算

本ベンチマークで構築した問題と参照回答群は所与のものとして扱い、評価対象のモデルの回答をすべての問題に対して生成し、その生成文と参照回答群間で複数の指標計算を行うことでベンチマークスコアを計測可能である。本章では、これらの 3 つのステップのすべてについて説明を行うが、実際にモデルの性能評価を行う際には、評価値計算のみを行えばよいことに注意されたい。

3.1 問題・回答例構築

問題の構築にあたって、日本語圏特有の常識として、日本の小中高の学習指導要領を参考に、下記の 13 科目から全 50 問の割り当てを決めた。

- 国語：4 問
- 社会（地歴、公民、環境[2)]）：各 4 問
- 算数：4 問
- 理科（生物、化学、物理、地学）：各 4 問
- 芸術、文化：各 4 問

2) 厳密には環境は学習指導要領上の科目として存在しないが、地歴・公民に分類できない問題が多いことから別の区分として設けた。

- 保健：4 問
- 情報：2 問

なお、数学は日本語圏特有のものが比較的少ないことから除外し、他の実技科目についても除外をしている。

これらの分類に基づいて、問題を 50 問作成した。例えば、化学の問題として、「接触法について教えて。」というように、比較的端的に回答できる問題を中心に構築した。また、人手により、回答例も構築を行った。回答例の構築にあたっては、公用文などの文体に近い 100 文字程度の文を目安として構築を行った。詳細な問題・回答例については、公開レポジトリを参照されたい。ここでは、最終的な 50 問の一覧は表 1 に示す。

3.2 参照回答群構築

前節で問題と回答例を人手で構築したものの、どの質問も答え方は一つではなく、様々な回答があり得ることから、参照用に参照回答群を構築する。この参照回答群は、日本語圏における、比較的尤度の高い回答を多く集めることで、回答分布を定義することを目的としている。

多くの回答を多くの日本語圏の日本語話者により人手で作成することが理想ではあるが、ここでは日本語特化かつ高性能な LLM で代替することとした。まず、日本語に特化という観点で行くと、ChatGPT [4] や GPT-4 [5]、LLaMa[1, 2] といったグローバルなモデルではなく、特に日本語コーパスからフルスクラッチ学習したモデルや追加事前学習により日本語性能を高めたモデルを選択することとした。さらに、高性能という観点では、70B や 100B といったパラメータが比較的大きいモデルを採用することにした。また、後述するが、参照回答群の生成においては、many-shots での生成を行うため、文体等のコントロールが比較的楽な事前学習モデル (指示学習前のモデル) を採用することとした。

これらの基準に従いかつモデルファミリーのバランスも考慮し、開発時点で利用可能であった、stockmark-100b[3]、PLaMo-100b、Swallow-MX-8x7b-NVE-v0.1[4] を採用した。

参照回答群を構築するにあたって、複数のステップで回答群を作成した。各ステップは以下のとおり

3） https://huggingface.co/stockmark/stockmark-100b
4） https://huggingface.co/tokyotech-llm/Swallow-MX-8x7b-NVE-v0.1

表 1 問題一覧

分類	問題
国語	競技かるたとは何ですか？
	漢文における返り点について教えて。
	擬音語と擬態語の違いは何ですか？
	重箱読みとは何ですか？
社会（地歴）	日本の開国について教えて。
	関ヶ原の戦いについて教えて。
	日本の東西南北端点について教えて。
	瀬戸内海式気候とは何ですか？
社会（公民）	天皇はどのような役割をもっていますか？
	三権分立とは何ですか？
	日本銀行の役割は何ですか？
	信用取引と先物取引の違いは何ですか？
社会（環境）	オゾン層って何ですか？
	再生可能エネルギーとは何ですか？
	四大公害病について教えて。
	夢の島の歴史について教えて。
算数	時計の長針と短針が 1 日に重なる回数は？
	つるかめ算について教えて。
	直角二等辺三角形の特徴を説明してください。
	算数と数学の違いは何ですか？
理科（生物）	ナメクジに塩をかけるとなぜ溶けてしまうの？
	ミドリムシの特徴を教えて。
	顕性と潜性の違いは？
	スズムシの鳴き声について教えて。
理科（化学）	タマネギを切ると涙が出るのはなぜ？
	接触法について教えて。
	温泉卵と半熟卵の違いは何から生まれるの？
	リトマス紙の使い方を教えて。
理科（物理）	ドップラー効果について教えて。
	超伝導とは何ですか？
	虹はどうして虹色なの？
	カミオカンデは何を行う施設ですか？
理科（地学）	日本はどうして地震が多いの？
	糸魚川静岡構造線とは何ですか？
	夏はどうして暑いの？
	地球の歴史について教えて。
芸術	天空の城ラピュタはどのような作品ですか？
	走れメロスはどのような作品ですか？
	山田耕筰は何をした人ですか？
	宝塚歌劇団の特徴は？
文化	春分の日と秋分の日はどのように決まるの？
	七草がゆについて教えて。
	神社と寺の違いについて教えて。
	神在月とは何ですか？
保健	日本脳炎とはどのような感染症ですか？
	柔道と合気道の違いを教えて。
	葛根湯とは何ですか？
	必須アミノ酸とは何ですか？
情報	Ruby について教えて。
	自然言語処理の主要な技術について教えて。

である。

- すべてのモデルで many-shots による各問 100 万回答を作成
- ルールベースでのハルシネーションの除去
- 頻度が極端に低い回答の除去を通じた稀なハルシネーションの除去
- 回答長と代表性を考慮に入れた各問 1000 回答への絞り込み

これらのフェーズは、代表的でかつ正確性の高い回答を幅広くとるための設計となっており、テイル事象を除去することに重きを置いている。以下では、さらに詳細にこれらのステップを説明する。

3.2.1 Step1: many-shots による各問 100 万回答の作成

できる限り人手で作成した回答例と同様に公用文に近い文体で 100 文字程度の回答を多く生成するために、many-shots による参照回答の生成を行った。明示的に文体と文字数を同等程度になるように指示をしたうえで、人手で作成したほかの 49 問から 20 問をランダムに抜き出し、20-shots での生成問題とした。下記に、実際に使用したプロンプトの例を示す。

Chat Completion の場合の例

{"role": "system", "content": "例と同様の文体及び文字数で、ユーザの質問に 1 行で答えてください。\n\n## 回答例 \nQ: 接触法について教えて。\nA: 接触法とは、硫黄を燃焼させて二酸化硫黄を作り、それを酸化バナジウム (V) の触媒を用いて酸化させて三酸化硫黄を作り、これを硫酸に吸収させて発煙硫酸とし、最後に希硫酸で希釈して濃硫酸を得る工業的製法です。\n..."}, {"role": "user", "content": "Q: 時計の長針と短針が 1 日に重なる回数は？"}

Completion の場合の例

例と同様の文体及び文字数で、ユーザの質問に 1 行で答えてください。

回答例
Q: 接触法について教えて
A: 接触法とは、硫黄を燃焼させて二酸化硫黄を作り、それを酸化バナジウム (V) の触媒を用いて酸化させて三酸化硫黄を作り、これを硫酸に吸収させて発煙硫酸とし、最後に希硫酸で希釈して濃硫酸を得る工業的製法です。
...
Q: 時計の長針と短針が 1 日に重なる回数は？
A:

なお、生成にあたっては、以下のパラメータを使用した。

- Temperature: 1.0
- Top-p: 0.98
- Top-k: 1000

また、外部情報がないと回答が難しい問題の場合、RAG を併用して回答を作成した。具体的には、「時計の長針と短針が 1 日に重なる回数は？」「重箱読みとは何ですか？」「接触法について教えて。」の 3 問において RAG を採用した。RAG のデータとしては Google の検索により発見した Web ページから各問 50 文ずつ適切な段落を抜き出し、得られた 50 文からランダムに選んだ 5 文程度を同時に与えるなどの工夫をすることで、特定の文をそのまま出力してしまわない形で文章を生成した。

これらの設定による参照回答群生成により、stockmark-100b、PLaMo-100b、Swallow-MX-8x7b-NVE-v0.1 の 3 つのモデルに対し、各問 100 万回答ずつ、合計 1.5 億回答を作成した。

また、公用文体の文章にするために、文化庁「公用文作成の考え方」（建議）[5] に準じて正規表現で文章の修正・削除を実施した。

3.2.2 Step2: ルールベースでのハルシネーションの除去

Step1 で生成した回答から、ハルシネーションを除去することとした。パラメータ数の多いモデルを採用してもなお、間違った回答をするケースは発生してしまうため、これらの除去は参照回答群の構築においては重要なプロセスである。

そこで、まず、Step2 としては、明確な間違いをルールベースで除去を行った。より具体的には、「時計の長針と短針が 1 日に重なる回数は？」という質問に対して、正しい回答が 22 であるのに対し、11 回や 23 回と答えてしまうようなケースを除くこ

5) https://www.bunka.go.jp/seisaku/bunkashingikai/kokugo/hokoku/pdf/93651301_01.pdf

とを目的としている。そのため、各問題への LLM の回答を目検で確認し、そういった誤りを正規表現等で NG 表現として定め、NG 表現にマッチする LLM の回答を除去した。

3.2.3 Step3: 頻度が極端に低い回答の除去を通じた稀なハルシネーションの除去

続いて、ハルシネーションの除去のもう一つの方法として、5-gram によるフィルタリングを実施した。

ルールベースでのハルシネーション除去は、LLM が答えを誤解しているケースや異なる知識を間違って学習してしまっている場合などに有効である一方で、偶然尤度の低い単語が紛れ込んでしまったりするなどして、通常ではありえない間違いをしてしまうケースなどを除去することが困難である。

そこで、Step3 では、文字レベルの n-gram の出現頻度に基づくフィルタリングを実施する。ここでは、文字レベルの 5-gram を採用し、各問の回答すべてを通じて 1 度しか出現しない 5-gram を含む文章を削除するというフィルタリングを採用した。これにより、「日本の東西南北端点について教えて。」という質問に対して「日本の東西南北端点は、東が南鳥島（東経 153° 59′ 30″ 、北緯 20° 43′ 57″ ）、…」と回答した例（正しくは東経 153° 59′ 12″ 、北緯 24° 16′ 59″ ）や、「関ヶ原の戦いについて教えて。」という質問に対して「関ヶ原の戦い（せっきがはらのたたかい）は、…」と回答した例（正しくは「せきがはらのたたかい」）などが除去された。大規模言語モデルで温度を下げて生成すると、見かけ上の性能は上がる一方で繰り返し文を多く生成するなど、人間が書く自然な文章とは異なる性質を持つ。そのため、フィルタリングを行うことで、不自然な文章が含まれないようにした。なお、フィルタリングにあたって、5-gram を用いた理由は、知識不足により答えたランダムな数値の羅列や稀に発生する誤字などを最もうまく除去できるできることが実験的に確認できたからである。

3.2.4 Step4: 回答長と代表性を考慮に入れた各問 1000 回答への絞り込み

次に、回答長を考慮に入れた参照回答群の選別を実施した。これは、100 文字程度の回答をターゲットにしたいという意図によるものである。そこで、回答字数が 100 文字に近い 3 万回答ずつ各問に対して抽出を行った。これにより、平均文字数 100 字（標準偏差 3.4 字）の参照回答群を獲得できた。

続いて、代表性を考慮に入れた最終的な各問 1000 回答への絞り込みを実施した。

3.3 モデルの評価値計算

前節で構築した参照回答群を用いて、評価対象のモデルのベンチマークスコアを計算する。

3.2.1 節で用いたプロンプトと同じプロンプトを使用し、評価対象のモデルの回答の生成を行う。ここで、生成の温度は特に指定はなく、また、生成回数についても自由である。温度を 0.0 からあげると生成結果にランダム性が発生する一方で、生成回数を増やすことで評価値の収束性を一定程度担保できる。そのため、評価値の試行回による分散が大きくなければ、どのような設定で生成しても構わない。実験的には、温度パラメータが大きすぎない限りは評価値の収束はよく、我々の実験では、Huggingface などのオープンなモデルでは温度 1.0、生成回数 100 回を採用した。一方で、OpenAI などの API を経由して利用するモデルの場合は、温度を 0.0 として、生成回数を 3 回程度に絞った。

続いて、生成した評価対象の回答に対して、前節で構築した参照回答群を用いて以下の 3 つの指標を計算する。

- Fluency: 文字レベルの 10-gram の出現割合の内積
- Truthfulness: 出現頻度 0.5%以上の文字レベルの 3-gram の割合
- Helpfulness: 手動で作成したルールによる評価

Fluency は参照回答群の平均評価スコアが 1.0 になるように各問ごとにスケールを行う。そのうえで、3 つのスコアの平均を取ることで最終的なベンチマークスコアとする。

以下では、これらの計算について詳しく説明する。

3.3.1 Fluency

Fluency とは、問題が与えられたときに、自然な日本語で回答できているかどうかを確認する指標であり、本研究で新たに定義する。

ここで、Fluency は、文字レベルの 1-gram から 10-gram のウィンドウに区切って文章を見たときに、参照回答群において尤度の高い文章になっている

かを確認する。そこで、評価対象の回答文からすべての文字レベルの 1-gram から 10-gram を抜き出し、それらの参照回答群における出現頻度を計算し、足しこんでいく。回答は 100 文字程度を想定しているが、出現頻度を足しこんでいく方式であると、文章長が長い場合にスコア上有利になってしまうため、100 文字から減衰を開始し、150 文字でスコアが 0 になる線形減衰によるスコアディスカウントを導入した。

つまり、評価対象の文が $C_1, C_2, \cdots, C_L$ という L 文字からなる文章であるとき、BOS および EOS を明示的に前後に追加し、$C_0, C_1, \cdots, C_L, C_{L+1}$ とする。文字レベルの w-gram は、$G_i^w = \{C_i, C_{i+1}, \cdots, C_{i+w-1}\}$ となり、$G_0^w, G_1^w, \cdots, G_{L-w+2}^w$ という $L-w+3$ 個の文字レベル 10-gram を構築できる。ここで、参照回答群における文字レベル w-gram の出現頻度を $L_{G_i^w}^w \in [0,1]$ と表記することしたときに、ディスカウント前スコアは、

$$F_w^\star := \sum_{i=0}^{L-w+2} L_{G_i^w}^w \tag{1}$$

となる。これは、評価対象の文章の文字レベル w-gram の頻度ベクトルと参照回答群の文字レベル w-gram の確率ベクトルの内積に相当する。

しかしながら、この $F_w^\star$ では、L の増加とともに単調増加してしまう。ゆえに、前述のディスカウントを導入した。その結果、w-gram に対応した Fluency Score は以下のように定義される。

$$F_w := \left(1 - \frac{\max(L-100, 0)}{50}\right) \sum_{i=0}^{L-w+2} L_{G_i^w}^w \tag{2}$$

$L = 150$ の際に、$F = 0$ となり、$L \leq 100$ の時に $F_w = F_w^\star$ となる。

さらに、この F_w を $w = 1, 2, \cdots, 10$ で計算して足すことで、最終的な Fluency score を計算できる。

$$F := \sum_{w=1}^{10} F_w \tag{3}$$

ここで、文字レベル w-gram の出現頻度 $L_{G_i^w}^w$ が平均的には同程度であると仮定をすると、

$$\sum_{i=0}^{L-w+2} L_{G_i^w}^w \sim L-w+3 \tag{4}$$

$$F_w \sim \left(1 - \frac{\max(L-100, 0)}{50}\right)(L-w+3) \tag{5}$$

と近似することが可能であり、$L = 100$ に単峰を持つ関数となることが確認できる。これが、全ての $w = 1, 2, \cdots, 10$ で成立するため、Fluency score$F$ も $L = 100$ に単峰を持つ関数となる。これにより、100 文字程度の回答をした場合に高いスコアになる特徴も持ち合わせている。

3.3.2 Truthfulness

Truthfulness とは、正確な情報をどの程度答えられているかどうかに関する指標として設計をした。3.2.3 節での考え方とほぼ同様に、参照回答群で極端に出現頻度の低い文字レベルの n-gram が出てきた場合にはハルシネーション等の正しくない回答が発生していると考える。Truthfulness では、極端な出現頻度の低さの閾値として 3-gram における 0.5%を採用し、0.5%以上の出現頻度の 3-gram の確率を計算することとした。ただし、ハルシネーションの有無を示す指標であることを鑑みて、句読点や鍵かっこなどの記号は頻度計算の対象から除外することした。さらに、Fluency と同様に 100 文字を超えた場合のディスカウントについても設定する。ただし、Truthfulness においては、100 文字を超えた場合については、それ以降をカットしてしまうことを認めることとする。つまり、100 文字を超えた場合については、Truthfulness スコアが最も高い文字数でカットをした場合のスコアを採用する。小さいモデルは文章を適切な文字数で出力することが難しい傾向があるが、ディスカウントを含めた最大値をとることで小さいモデルの出力のランダム性を吸収し、スコアを安定させる効果があるため、このような設計とした。

前節と同様のノーテーションで数式的に定義する。文字レベルの 3-gram は、$G_i^3 = \{C_i, C_{i+1}, C_{i+2}\}$ となる。また、参照回答群における文字レベル 3-gram の出現頻度を $L_{G_i^3}^3 \in [0,1]$ と表記する。さらに、$C_1, C_2, \cdots, C_L$ という L 文字からなる文章のうち、句読点や記号に該当しない文字のインデックスを $\Omega_L = \{i | C_i \notin \{$ 句読点、記号 $\}\} \subset \{1, 2, \cdots, L\}$ と表記する。ただし、前述の通り、100 文字以上の回答の場合には、出力をカットして Truthfulness を計算することを認めるため、$\Omega_I = \{i | C_i \notin \{$ 句読点、記号 $\}\} \subset \{1, 2, \cdots, I\}$ (ただし $100 \leq I \leq L$ または $I = L$) とする。

このとき、ディスカウント前の Truthfulness は以

下の通り計算される。

$$T_I^\star := \frac{1}{|\Omega_I|} \sum_{i \in \Omega_I} \frac{\min\left(L_i^{3\star}, 0.005\right)}{0.0005} \tag{6}$$

$$L_i^{3\star} := \max\left(L^3_{G_{i-1}^3}, L^3_{G_i^3}, L^3_{G_{i+1}^3}\right) \tag{7}$$

しかしながら、この $T_I^\star$ は、$F^\star$ と同様に I の増加とともに $|\Omega_I|$ が大きくなるため、単調増加してしまう。ゆえに、同様にディスカウントを導入する。その結果、出力長 I に対する暫定 Truthfulness Score は以下のように定義される。

$$T_I := \left(1 - \frac{\max(I - 100, 0)}{50}\right) T_I^\star \tag{8}$$

さらに、I 文字にカットアウトした最大の Truthfulness Score を最終的な値とするため、最終的な Truthfulness Score は以下のように計算される。

$$T := \begin{cases} \left(1 - \frac{\max(L-100,0)}{50}\right) T_L^\star & (L \leq 100) \\ \max\limits_{I \in \{100,101,\cdots,L\}} \left(1 - \frac{\max(I-100,0)}{50}\right) T_I^\star & (100 \leq L) \end{cases} \tag{9}$$

$|\Omega_I|$ が概ね I と一致することから、Fluency の場合と同様に、T_I はおおむね $I = 100$ の時に最大値を取る。そのため、この指標は、概ね 100 文字までにおける出現頻度が極端に低くない 3-gram の比率を示しているものと解釈可能であり、冒頭 100 文字にハルシネーションが入らない確率を見ているものとして理解できる。また、Fluency と異なり、出力長に依存しない評価値となっている。

3.3.3 Helpfulness

Helpfulness は、評価対象の文章がどれだけ必要情報を適切に含んでいるかを評価する、人手で構築したルールベースの指標である。各問ごとに、含むべき重要単語を定義し、それがどれだけ含まれているかを割合を計測する。例えば、「つるかめ算について教えて。」という質問に対して、「合計」、「それぞれ」、「各々」の 3 つのうち 1 つの単語と「算数」の計 2 単語を必須重要単語として設定している。このように、and/or を用いて必須単語の条件を定義している。なお、原則として、これらの必須重要単語は等価に扱うこととしているが、重みづけを変えるケースも存在する。ただし、100 文字を超えている場合については、100 文字までで打ち切った場合の評価値と、$I > 100$ となる I 文字目までで打ち切った場合の評価値に $\left(1 - \frac{\max(I-100,0)}{50}\right)$ を乗じた値をすべて計算し、最も高い値を Helpfulness score として採用する。これにより、文字数オーバー部分のディスカウントしてでも加算したほうがよい単語を考慮に入れることができるため、安定的な評価ができる。

4 実験

3 章の提案ベンチマークの有効性を確認するために、複数の実験を行う。まず、3 章で明確ではないことして、2 点あげられる。1 点目は提案したベンチマークの計算方法が LLM の性能を計るにあたって、有効な手法となっているかという点である。2 点目は 50 問の問題集の問題としての有効性であり、真にバランスよく LLM の性能を計ることのできる問題となっているかということが疑問として残る。

1 点目に対応する実験として、本研究で提案したベンチマークが従来手法と相関性があるかどうかを検証する。具体的には、提案ベンチマークと LLM-as-a-judge[30] による評価の相関性を確認する。

2 点目に対応する実験として、生成の良さを確認する従来指標である Japanese MT-bench との相関性を確認する。これにより、本ベンチマークで構築した問題集が、MT-bench におけるどのような特性と一致しているのかを確認することができるため、本ベンチマークが見ている性能を検証可能であると考えられる。

以下では、予備実験的なベンチマークの特性を見る実験と、これらの 2 点に対応する実験について詳しく説明する。

4.1 実験 1: 提案ベンチマークの特性分析

まず、提案ベンチマークが適切に機能しているかを確認するために、その特性を検証するような予備実験を行う。

ここでは、Fluency, Truthfulness, Helpfulness についてそれぞれのモデルのスコアをプロットすることで、それらの関係性について確認する。

加えて、提案手法における、参照回答群の安定性について確認する。参照回答群の構築にあたっては、開発時点で利用可能であった、stockmark-100、PLaMo-100b、Swallow-MX-8x7b-NVE-v0.1 の 3 つのモデルを利用して、それらをアンサンブルする形で用いたが、これらのモデル間でのスコアのばらつきの有無を確認するために、アンサンブルせずに 1 つのモデルのみを用いた場合に、ベンチマークの差が出るかどうかを検証する。

これらの検証を通じて、提案ベンチマークの特性

を明確にする。

4.2 実験 2: LLM-as-a-judge との比較

前述の通り、提案ベンチマークと LLM-as-a-judge[30] による評価の相関性を確認する。ここでは、本ベンチマークを開発時点で利用可能であった、多数の主要な LLM を用いて、それらの LLM の回答に対する提案手法のスコアリングと、LLM-as-a-judge [30] として、OpenAI 社の GPT-4o (モデルバージョン: 2024-05-13) を用いて 10 段階評価を行った場合の score の比較を行った。なお、LLM が chat Completion と Completion の両方に対応している場合は、その両方での生成結果の評価を対象とした。

最終的には相関係数を見ることで、その有効性の確認を行う。なお、ここでは、LLM-as-a-judge を妥当な評価指標とみなしており、LLM のベンチマーク計測においては LLM-as-a-judge での計測で充分ではないかと考えることもできるが、実際には、LLM-as-a-judge は大規模で高性能な LLM を非常に多くの量回す必要があるため、API の利用費やオンプレミスで実行した場合でも計算コストが膨大にかかる問題があることに注意されたい。つまり、提案手法では、非常に低コストでベンチマーク計測ができるということである。

なお、この比較実験において、LLM-as-a-judge のプロンプトは以下のものを利用した。

LLM-as-a-judge のプロンプト

[Instruction]
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider correctness and helpfulness. You will be given a reference answer and the assistant's answer. Begin your evaluation by comparing the assistant's answer with the reference answer. Identify and correct any mistakes. Be as objective as possible. The expected language is Japanese. Responses in languages other than Japanese will incur score deductions unless specifically required. Failure to use Japanese at all will result in the lowest evaluation. However, using Japanese is not mandatory when providing only Python scripts or calculation results, where Japanese is not essential. Additionally, your explanation of judgement should be in Japanese. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[[rating]]", for example: "Rating: [[5]]".

[Question]
{{ 質問文 }}

[The Start of Reference Answer]
{{ 想定回答 }}
[The End of Reference Answer]

[The Start of Assistant's Answer]
{{ モデルの出力 }}
[The End of Assistant's Answer]

4.3 実験 3: 既存ベンチマークとの比較

続いて、本ベンチマークが従来手法のベンチマークと整合性が取れているかについて検証を行う。

ここでは、主に Japanese MT-bench を用いて、それとの相関を確認することにより、提案手法が評価手法として妥当性があるかどうかについて検証を行う。比較にあたっては、Nejumi LLM リーダーボード 3[6] に掲載のスコアを用いることとする。そのため、本研究においては、MT-bench との相関だけではなく、Nejumi LLM リーダーボード 3 の総合点との相関性についても確認を行った。

なお、相関性の評価にあたっては、対象の LLM が Chat Completion と Completion の両方に対応している場合には、1 つのモデルに対して提案手法におけるスコアが 2 つ存在しているため、Chat Completion と Completion の両方の提案手法でのスコアのうち、高い方を相関の計算に使用することとした。

実験 1 で対象としたモデルと Nejumi LLM リーダーボード 3 が対象にしている LLM に差異が存在するため、それらの両方で評価されている 37 モデルを対象に評価を行った。

6) https://wandb.ai/wandb-japan/llm-leaderboard3/reports/Nejumi-LLM-3--Vmlldzo3OTg2NjM2

5　結果

5.1　実験 1: 提案ベンチマークの特性分析

まず、提案ベンチマークが適切に機能しているかを確認するために、その特性を検証する。図 1–3 に、Fluency, Truthfulness, Helpfulness の関係性についてプロットした。また、表 2 に主要なモデルのスコアも示す。

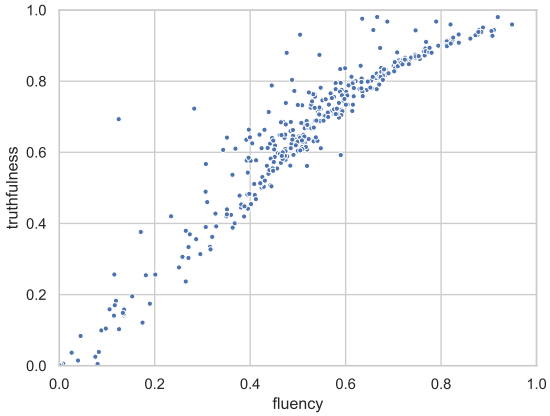

図 1　Fluency と Truthfulness の関係

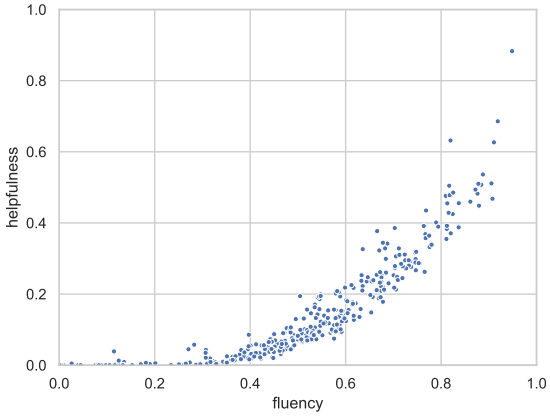

図 2　Fluency と Helpfulness の関係

この図によると、Truthfulness、Fluency、Helpfulness の順に容易なタスクであることがわかる。図 2 と 3 を確認すると、Helpful のスコアは Fluency と Truthfulness のスコアが 0.6 0.8 程度までかなり高くなってきて初めて数値が 0.2 を超えてくることが確認できるため、Helpful が最もスコアを上げにくいタスクであると言える。また、図 1 を確認すると、Truthfulness の方が若干スコアの立ち上がりが早く、より簡単であるタスクであると言える。この関係性

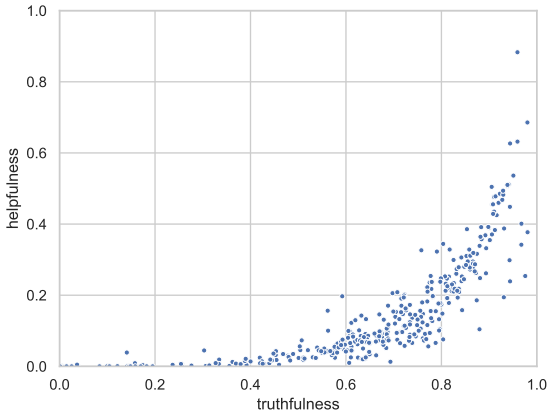

図 3　Truthfulness と Helpfulness の関係

は、図 2 と 3 において、Helpfulness が 0.2 となる点が Fluency は約 0.6、Truthfulness が約 0.8 であることからも確認できる。

一方で、それぞれのモデル間での順序性は、多少の揺らぎがあるものの、どの指標においても概ね一貫しているようにも見える。また、Fluency、Truthfulness に関しては、現状ですでに 1.0 にちかいスコアを達成しているモデルが存在しており、限界が近い一方で、Helpfulness はまだまだ 1.0 に近いスコアを達成できていないものが多い。

続いて、提案手法における、参照回答群の安定性について確認する。参照回答群の構築にあたっては、開発時点で利用可能であった、stockmark-100、PLaMo-100b、Swallow-MX-8x7b-NVE-v0.1 の 3 つのモデルを利用して、それらをアンサンブルする形で用いたが、これらのモデル間でのスコアのばらつきの有無を確認するために、アンサンブルせずに 1 つのモデルのみを用いた場合に、ベンチマークの差が出るかどうかを検証する。

図 4–6 に結果を示したが、どのモデルを使用した場合においても、ほとんど性能の差がなく、0.999 を超える相関が確認できた。これにより、参照回答群は比較的安定しており、参照回答群の作成に使用したモデルの依存というのは高くないということが分かった。

5.2　実験 2: LLM-as-a-judge との比較

続いて、提案ベンチマークと GPT-4o による LLM-as-a-judge の比較を行う。

ここでは、LLM-as-a-judge に GPT-4o の API の利用金額が大幅にかかることから、前節のモデルから

表 2　主要なモデルのスコア一覧。全ての結果は https://github.com/pfnet-research/pfgen-bench を参照されたい。

model	type	Score	Fluency	Truthfulness	Helpfulness
回答例 (人間)	N/A	1.0501	1.155	0.996	1.000
anthropic/claude-3-5-sonnet-20240620	chat	0.9303	0.949	0.959	0.883
openai/gpt-4o	chat	0.8615	0.919	0.980	0.686
openai/gpt-4	chat	0.7916	0.888	0.951	0.536
tokyotech-llm/Swallow-70b-NVE-instruct-hf	completion	0.7766	0.884	0.938	0.507
pfnet/plamo-100b	completion	0.7469	0.861	0.920	0.460
CohereForAI/c4ai-command-r-plus	completion	0.7365	0.818	0.913	0.478
nvidia/nemotron-4-340b-instruct	completion	0.7175	0.816	0.908	0.429
meta-llama/Meta-Llama-3.1-405B	completion	0.6759	0.767	0.892	0.368
google/gemini-1.5-pro-001	chat	0.6745	0.666	0.980	0.377
stabilityai/japanese-stablelm-base-beta-70b	completion	0.6202	0.733	0.848	0.280
openai/gpt-35-turbo	chat	0.6136	0.658	0.944	0.239
meta-llama/Meta-Llama-3.1-70B	completion	0.5659	0.665	0.822	0.211
Qwen/Qwen-72B-Chat	completion	0.5002	0.614	0.716	0.171
mistralai/Mixtral-8x22B-v0.1	completion	0.4050	0.517	0.615	0.084
mistralai/Mixtral-8x7B-Instruct-v0.1	completion	0.3914	0.488	0.636	0.050

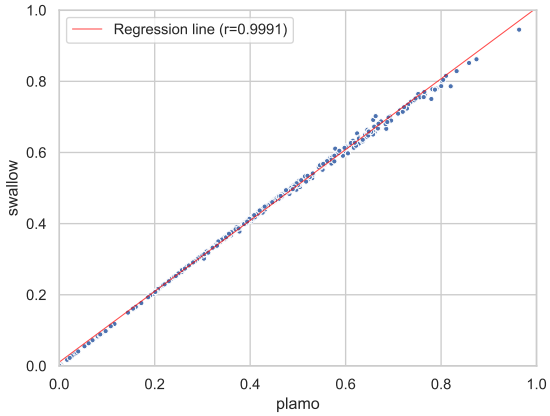

図 4　参照回答群の構築に PLaMo-100b を使った場合と Swallow-MX-8x7b-NVE-v0.1 を使った場合のスコア比較

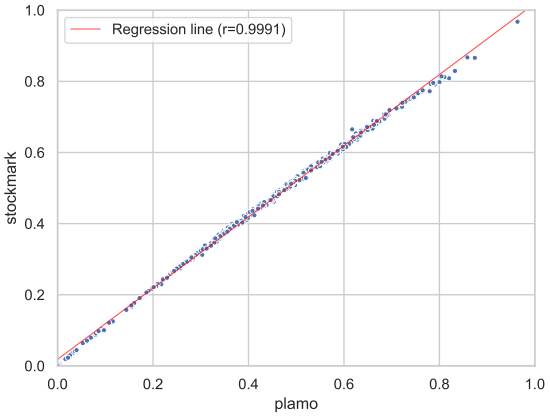

図 5　参照回答群の構築に PLaMo-100b を使った場合と stockmark-100 を使った場合のスコア比較

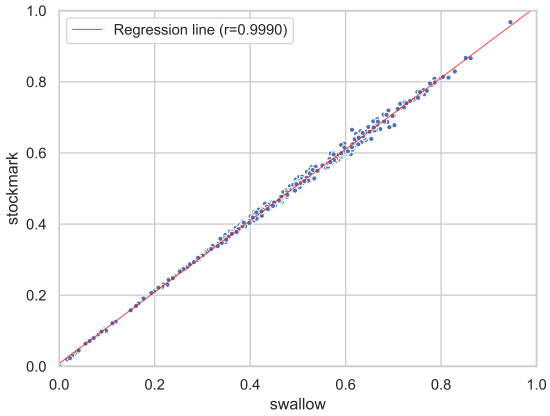

図 6　照回答群の構築に Swallow-MX-8x7b-NVE-v0.1 を使った場合と stockmark-100 を使った場合のスコア比較

さらに主要なものに絞った 50 個のモデルに対して比較を行った。

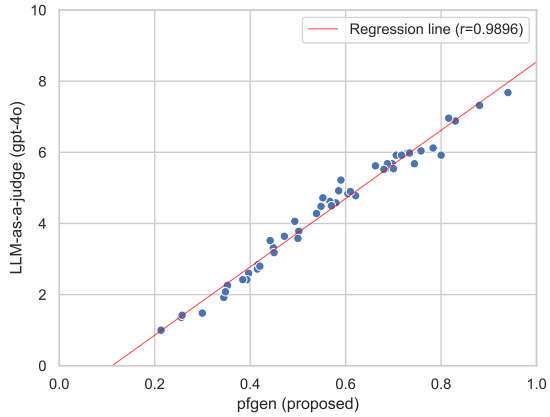

図 7　提案ベンチマークと GPT-4o による LLM=as-a-judge のスコアの比較

図 7 にその結果を示す。結果として、0.9896 というとても高い相関性を示しており、ベンチマークとしては充分に機能していると言える。今回の実験では、提案手法におけるスコアが 1.0 を上回る領域 (GPT-4o による LLM-as-a-judge で 9 を上回る領域) や、0.2 を下回る領域 (GPT-4o による LLM-as-a-judge で 0 点付近の領域) でのスコアの挙動を観測できていないものの、現状の実用のレベルでは既存手法である GPT-4o による評価と同等程度に分解能を持っており、充分であると言えるのではないだろうか。

5.3　実験 3: 既存ベンチマークとの比較

続いて、Nejumi LLM リーダーボード 3 をもちいて、Japanese MT-bench と Nejumi Bench との相関を確認する。

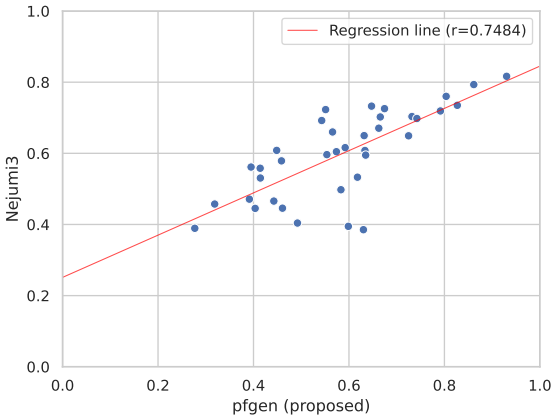

図 8　提案ベンチマークと Nejumi LLM リーダーボード 3 のスコアの比較

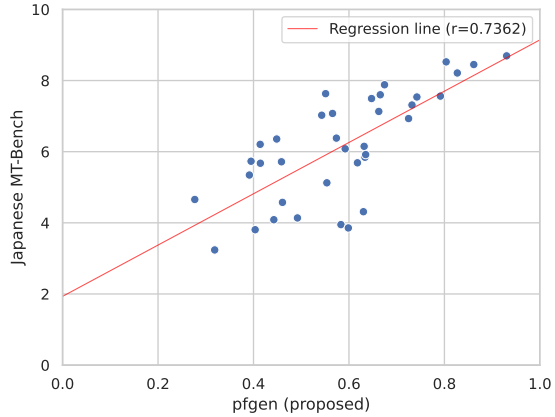

図 9　提案ベンチマークと Japanese MT-bench のスコアの比較

図 8, 9 に結果を示す。まず、どちらのベンチマークとも、0.7 を上回る比較的高い相関性があることが確認できた。一方で、前節の LLM-as-a-judge との比較と比べると、少し低い相関となり、ばらつき具合も大きくなっていることが確認できた。

これはタスクの違い由来のものであると考えることができるが、必ずしもどれが正解ということもないため、判断が難しい。しかしながら、特に上位のパフォーマンスを示す LLM の性能評価については、比較的 Nejumi LLM リーダーボード 3 と Japanese MT-bench と提案ベンチマークでは比較的一貫した評価となっており、提案ベンチマークが充分にベンチマークとして機能しているのではないかと考えることができる。

また、0.74 という相関性も決して低いものではなく、LLM の性能評価として提案ベンチマークの有効性が一定レベルで示されているともいえる。

6 考察

実験の結果、提案ベンチマークは一定のレベルで有効なベンチマークとして一定レベルで機能していることを確認できた。特に LLM-as-a-judge との相関性の確認は非常に高い相関を示しており、LLM-as-a-judge という計算コストが大きい手法を代替できる可能性が示唆されていると考えられる。

ここでは、なぜこのベンチマークが有効であるかどうかという点について議論する。

まず、Helpfulness に関しては、人手でルールを構築しているため、常識をどの程度学習できているのかどうかを計る指標となっていると言える。一方で、Fluency と Truthfulness は、参照回答群との乖離度を測る指標になっていると解釈できると考えられる。

その場合、Fluency と Truthfulness が十分に機能するためには、参照回答群の性能に依存するようにも見えてしまうが、今回の結果を分析すると、参照回答群を作成する際に使った LLM や参照回答群自体のベンチマークスコアを上回るスコアを出しているモデルも存在した。具体的には、anthropic/claude-3-5-sonnet-20240620 と openai/gpt-4o が、参照回答群をモデルの回答としてベンチマークを計測した場合よりも良い性能を発揮していた。個別の指標で見ても、anthropic/claude-3-5-sonnet-20240620 は Fluency で、openai/gpt-4o は Truthfulness で参照回答群のスコアを上回っていた。つまり、今回の提案手法は、参照回答群の生成結果の性能以上のものを分析できる可能性を示唆している。

ここで、LLM-as-a-judge が A という回答に対して J 点 ($J = 0, 1, \cdots, 10$) と評価する確率を $L_j(Q, A, J)$ と表記する。ここで、質問 Q に対する良い回答というのは、$\{A | \arg\max_J L_j(Q, A, J) = 10\}$ と定義可能である。さらに、LLM-as-a-judge が理想的に機能しているとすると、$\{A | \arg\max_J L_j(Q, A, J) = 9\}$ は $\{A | \arg\max_J L_j(Q, A, J) = 10\}$ の隣接空間になると考えることができる。ここで、$\{A | \arg\max_J L_j(Q, A, J) = i\}$ が $\{A | \arg\max_J L_j(Q, A, J) = i + 1\}$ に隣接する空間であると仮定すると、$\{A | \arg\max_J L_j(Q, A, J) = 10\}$ が取得可能であり、二つの回答間のよさの距離を理想的に計算できる指標が存在するとすると、良い回答群 $\{A | \arg\max_J L_j(Q, A, J) = 10\}$ からの距離指標だけで生成の良さを定義できるうえ、LLM-as-a-judge よりも良い評価を行うことができることを意味する。今回の実験では、LLM-as-a-judge 並みの性能を持つベンチマークを構築できているため、参照回答群が十分によい回答群として機能しており、また、指標計算方法も十分に良い距離指標となっている可能性が示唆される。

さらに、今回のベンチマークで使用した手法は、ほぼすべてで n-gram だけに着目している。Fluency は 1–10 gram、Truthfulness は 1–3 gram、Helpfulness は単語の存在なので 1 gram 程度となっている。これらを考慮に入れると、質問への回答の良さを計るにあたっては、n-gram という特徴量程度で充分である可能性が考えられる。そもそも文法に従えば、ある程度次に来る単語というものは制約されており、n-gram のありうる空間 (V^n; ここで、V は語彙サイズ) からすると、非常に限られた空間であると言える。そのうえで、さらに、特定の質問への回答である条件付けを行うことで、その許容空間というものはさらに絞られると考えられる。それゆえに n-gram を見るだけでも、生成の良さを評価できる可能性を今回の実験は示唆している。加えて、今回は 100 文字制限をしたため、10-gram は特徴量としてかなり強いと考えられるが、3-gram までしか使っていない Truthfulness だけでも性能指標としては機能していることを考慮すると、日本語の生成は、直前 2 文字だけでも相当次の文字が絞られるということを意味しているともいえる。これらの考察は、LLM でランダムに生成させた n-gram を見るだけで LLM の性

能を評価できる指標の構築可能性を示しているとも考えられるため、それらの検証および理論的な解析は今後の課題である。

7 まとめ

本研究では、LLM の対話における生成の良さを計るベンチマークとして、pfgen-bench を提案した。このベンチマークは、日本語圏特有の常識を問う 13 科目 50 問の問題からなり、評価対象の LLM がこれらの問題に対する回答を 100 文字程度で作成したのちに、スコアリングが行われる。スコアリングにあたっては、事前に構築された参照回答群を用いて、様々な LLM に対するベンチマークに対する Fluency、Truthfulness の 2 指標を n-gram に着目して計算し、さらに、人手で作成したルールベースのアルゴリズムからなる Helpfulness を組み合わせて計算される。参照回答群の構築にあたっては、主に 3 つの日本語の大規模言語モデルを用いて構築を行った。実験の結果、提案ベンチマークは、GPT-4o を用いた LLM-as-a-judge とほぼ同様の評価が行えることが確認でき、既存の Japanese MT-bench などのベンチマークとも高い相関性を持つことが確認できた。

謝辞

データセットを作成するにあたり、片岡俊基氏に回答の正しさについて校閲を行っていただきました。

Declarations

著者らは、pfnet/plamo-100b の開発元である、株式会社 Preferred Networks/Elements に所属しているが、本研究におけるモデル選定などにおいては、客観的根拠を以って公平な評価を行うように努めている。また、透明性の確保のために、ベンチマークの計測コードを公開している。

参考文献

[1] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and Efficient Foundation Language Models. **arXiv**, 2023. https://arxiv.org/abs/2302.13971.

[2] Hugo Touvron, Louis Martin, et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. **arXiv**, 2023. https://arxiv.org/abs/2307.09288v2.

[3] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. **arXiv preprint arXiv:2310.06825**, 2023.

[4] OpenAI. ChatGPT, 2023. https://openai.com/blog/chatgpt/.

[5] OpenAI. GPT-4 Technical Report, 2023.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In **Advances in Neural Information Processing Systems**, Vol. 30, pp. 5999–6009, 2017.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics**, pp. 4171–4186. Association for Computational Linguistics, 2019.

[8] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training, 2018. https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[9] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners, 2019. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[10] Tom Brown, Benjamin Mann, et al. Language Models are Few-Shot Learners. **Advances in Neural Information Processing Systems**, Vol. 33, pp. 1877–1901, 2020.

[11] Google. Bard, 2023. https://bard.google.com/.

[12] Databricks. Dolly, 2023. https://github.com/databrickslabs/dolly.

[13] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. **arXiv**, 2022. https://arxiv.org/abs/2211.05100.

[14] Vicuna. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality, 2023. https://vicuna.lmsys.org/.

[15] Aakanksha Chowdhery, Sharan Narang, et al. PaLM: Scaling Language Modeling with Pathways. **arXiv**, 2022. https://arxiv.org/abs/2204.02311v5.

[16] Rohan Anil, Andrew M. Dai, et al. PaLM 2 Technical Report. **arXiv**, 2023. https://arxiv.org/abs/2305.10403v3.

[17] Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, et al. A framework for few-shot language model evaluation, 2021. https://github.com/EleutherAI/lm-evaluation-harness.

[18] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. **Proceedings of the International Conference on Learning Representations (ICLR)**, 2021.

[19] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni,

Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. **arXiv preprint arXiv:2406.01574**, 2024.

[20] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. **arXiv preprint arXiv:2311.12022**, 2023.

[21] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. **arXiv preprint arXiv:2110.14168**, 2021.

[22] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. **NeurIPS**, 2021.

[23] Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. Language models are multilingual chain-of-thought reasoners. **arXiv preprint arXiv:2210.03057**, 2022.

[24] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. 2021.

[25] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. **arXiv preprint arXiv:2107.03374**, 2021.

[26] Marc Eulerich, Aida Sanatizadeh, Hamid Vakilzadeh, and David A. Wood. Is it All Hype? ChatGPT' s Performance and Disruptive Potential in the Accounting and Auditing Industries. **SSRN Electronic Journal**, 2023. https://papers.ssrn.com/abstract=4452175.

[27] Harsha Nori, Nicholas King, Scott Mayer Mckinney, Dean Carignan, and Eric Horvitz. Capabilities of GPT-4 on Medical Challenge Problems. **arXiv**, 2023. https://arxiv.org/abs/2303.13375v2.

[28] Kwan Yuen Iu and Vanessa Man-Yi Wong. ChatGPT by OpenAI: The End of Litigation Lawyers? **SSRN Electronic Journal**, 2023. https://papers.ssrn.com/abstract=4339839.

[29] Jonathan H. Choi, Kristin E. Hickman, Amy Monahan, and Daniel B. Schwarcz. ChatGPT Goes to Law School. **SSRN Electronic Journal**, 2023. https://papers.ssrn.com/abstract=4335905.

[30] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, **Advances in Neural Information Processing Systems**, Vol. 36, pp. 46595–46623. Curran Associates, Inc., 2023.

[31] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. **arXiv preprint arXiv:2403.04132**, 2024.