Swallow コーパス v2: 教育的な日本語ウェブコーパスの構築

服部 翔 [1,2]　岡崎 直観 [1,2,3]　水木 栄 [1,2]　藤井 一喜 [1,2]　中村 泰士 [1,2]
大井 聖也 [1,2]　塩谷 泰平 [1]　齋藤 幸史郎 [1]　Youmi Ma[1]　前田 航希 [1]
岡本 拓己 [1]　石田 茂樹 [1]　横田 理央 [1,2,3]　高村 大也 [2]
[1] 東京科学大学　[2] 産業技術総合研究所　[3] NII LLMC
{kakeru.hattori@nlp., okazaki@, swallow@nlp.}comp.isct.ac.jp

概要

大規模言語モデル（LLM）の事前学習では，高品質なテキストを用いることが望ましい．本研究では，文書の「教育的価値」に着目した 2 種類の軽量な分類器を構築して，各文書に品質スコアを付与し，大規模日本語ウェブコーパスから高品質なテキストを抽出する手法を提案する．実験により，提案手法を適用することで，同等の学習計算規模で日本語の知識に関する LLM の能力をより効率的に向上できることを示した．また，分類器の特性比較，ヒューリスティック・ルールの調整，学習のエポック数を増やす実験などを通じて，提案手法の実用性や LLM 構築の最良慣行について検証する．

1　はじめに

LLM の性能はスケーリング則 [1] に従い，パラメータ数，計算予算，訓練データ量との間にべき乗則が成り立つと言われている．ところが，単に数量を増やせば LLM が高性能になるのではなく，適切なアーキテクチャの選択や高品質な訓練データの準備が欠かせない．LLM の訓練データとして，Common Crawl[1] と呼ばれるウェブアーカイブを活用することが多く，英語では RefinedWeb [2] や RedPajama [3]，日本語では Okazaki ら [4]，新里ら [5]，榎本ら [6]，Tolmachev ら [7] 等の構築例がある．

最近では，学習に用いるウェブ文書を厳選すると，同じ訓練データ量でも高い性能を達成できると報告されている．Penedo ら [8] は学校のカリキュラムへの関連性や内容の有益性を評価基準とした「教育スコア」を付与する分類器を構築し，ウェブ文書をスコア付けすることで，LLM にとって「教育的価値」の高い訓練データ（FineWeb-Edu）を構築した．また，Li ら [9] は OpenHermes2.5 [10] や ELI5[2] を正例として訓練した分類器を用い，高品質な英語コーパス（DCLM-baseline）を構築した．

本研究では，LLM の知識や一般教養を高めることを「教育」の主眼と捉え，教育的価値の高いテキストを厳選した Swallow コーパス v2 を構築する．具体的には，文書の教育的価値を判定する 2 種類の分類器を構築し，日本語ウェブコーパスから教育的価値の高い文書を厳選する手法を提案する．また，構築したコーパスで LLM の継続事前学習を行い，提案手法の有効性や LLM 構築の最良慣行を多角的に検証する．本研究で得た知見は以下の通りである．

1. 提案手法は主に，日本語の知識系タスク（QA，教養科目，翻訳）の性能改善に有効である．
2. 教育的価値の分類器の訓練データとして，特定の文書（Wikipedia）を正例とするよりも，LLM に教育的価値を採点させた文書を用いるほうが，より広範な文書に適切なスコアを付与できるため，汎用性・有効性が高い．
3. Swallow コーパス v1 [4] で採用されていたフィルタリング・ルールよりも，提案手法の方がより LLM の知識や一般教養を高められる．
4. 学習トークン数の増加を狙い，エポック数を増やしても提案手法の有効性は変わらないが，過度にエポック数を増やすと性能が低下する．

2　Swallow コーパス v2 の構築

Swallow コーパス v2 は，Common Crawl から日本語テキストを抽出し，重複除去を済ませてから提案手法で品質フィルタリングを行い，LLM にとって教育的価値の高い文書を厳選するという手順で構築される．日本語テキストの抽出[3] や重複除去[4] の実

1）https://commoncrawl.org/

2）専門知識を一般人に分かりやすく説明するための掲示板．
3）https://github.com/swallow-llm/swallow-corpus
4）https://github.com/swallow-llm/doubri

装は Swallow コーパス v1 [4] の時と同一であるが，v2 では Common Crawl の利用範囲を拡大し，2013 年から 2023 年末までの 94 アーカイブを用いた．また，v1 では重複除去を品質フィルタリング後に行っていたが，これを先に済ませることで，品質フィルタリングの手法を後から試行錯誤できるようになった．重複除去後のコーパスの規模は約 3.2 兆文字（約 19 億ページ）である．

品質フィルタリングでは，ヒューリスティックルールに基づく手法（付録・表 4）を適用してから，提案する分類器（3 節）を適用する．ヒューリスティックルールは，v1 で設計したルールを緩和し，有益な文書を過度に除去することを防いだ．具体的には，(1) 文の平均文字数，(2) 非日本語文字の割合，(3) 複数回登場する 5〜10gram の割合などのルールを取りやめ，論文などの学術的な文書や英語教材などの多言語文書を除去しすぎないように調整した．

3 教育的価値の分類器

本研究では先行研究 [8, 9] を参考に，(1) 内容が学術的・教養的である，(2) 物事を分かりやすく教えている，の 2 点を満たす文書が「教育的」であると考える．そして，(1) 学術分野の Wikipedia 記事，(2) 教育的価値を LLM に自動採点させたウェブ文書を訓練データとして，2 種類の分類器を fastText [11] で学習した．本稿では前者を **Wiki 分類器**，後者を **LLM 分類器**と呼ぶ．いずれの分類器も文字 n-gram（$n = 2, 3$）を特徴量とした．fastText は CPU で高速に動作し，大量の文書を低コストで処理できる．構築した分類器は HuggingFace 上で公開[5] している．

3.1 Wiki 分類器

従来研究では，Wikipedia を高品質なテキストのお手本として用いることがある [5, 9, 12]．そこで，本研究では Wikipedia 記事を教育的な文書の正例と見なし，分類器を構築した．人物に関する記事など，必ずしも教養的とは言えない記事もあるため，学術分野のカテゴリ[6] に属する日本語 Wikipedia 記事 37,399 件を抽出し，訓練データの正例とした．また，負例は Swallow コーパス v2 からランダムにサンプリングした文書 37,399 件とした．Wiki 分類器を fastText の二値分類器として訓練し，正例の予測確率（0〜1）を文書の教育的スコアとした．

5） https://huggingface.co/tokyotech-llm/edu-classifier
6） https://ja.wikipedia.org/wiki/学問の一覧 をもとに，ヒューリスティクスで 2,000 件のカテゴリを選択した．

表 1 Wiki 分類器と LLM 分類器の精度

セット	Wiki	LLM			
	Acc	4-Acc	RMSE	MAE	2-Acc
訓練	0.998	0.908	0.334	0.209	0.960
評価	0.995	0.667	0.565	0.399	0.899

3.2 LLM 分類器

FineWeb-Edu [8] を参考に，ウェブ文書に教育的スコアを自動付与した．具体的には，Swallow コーパスから 20 万件の文書をランダムに抽出し，さらに独自に選定したウェブ記事 31,059 件を加え，訓練データとした．Llama 3.1 70B Instruct にプロンプト（付録・図 3）を与え，「高度に学術的なトピックか」「深い洞察や議論を提供しているか」「一般向けに分かりやすいか」の 3 つの基準で文書を自動採点した．各基準が 1 点の配点を持つこととし，3 点満点の加点方式のスコアを算出するように LLM に指示した．自動採点された文書を訓練データとして，LLM 分類器を fastText で訓練した[7]．LLM 分類器は教育的スコアの点数（0,1,2,3）を予測する 4 クラス分類器として訓練し，各ラベルの予測確率に基づくスコアの期待値（0〜3）を文書の教育的スコアとした．

3.3 分類器の評価と適用

Wiki 分類器と LLM 分類器の精度を表 1 に示す．Wiki 分類器は評価セットで 99%以上の正解率（Acc）を達成した．LLM 分類器は 4 クラス分類であるため，二値分類よりも難易度が高く，ラベル予測の正解率（4-Acc）そのものは低いが，二乗平均平方根誤差（RMSE）や平均絶対誤差（MAE），スコア 1.5 を閾値とした二値分類の正解率（2-Acc）は良好で，十分な性能であると判断した．

Swallow コーパス v2 全体にこれらの分類器を適用し，得られた教育的スコアの分布を図 1 に描いた．Wiki 分類器は算出されるスコアが 0 付近に偏っており，コーパス中に Wikipedia 記事と類似する文書があまり含まれていないことを反映している．一方，LLM 分類器はスコア分布のピークが 1 付近にあり，サンプルデータに LLM が直接付与したスコア分布（表 5）ともほぼ一致している．スコアが 1.5 以上の文書は全体の約 15%であり，教育的価値が高いと判断される文書の割合はあまり高くない．

7） LLM 分類器という名前ではあるが，LLM の役割は全てのウェブ文書を分類することではなく，fastText 分類器の訓練データを作成することである．

表 2　継続事前学習前後の日英ベンチマークスコア比較

実験設定		QA JCom.	QA JEMHopQA	QA NIILC	教養科目 JMMLU	英日翻訳 WMT20	日英翻訳 WMT20	要約 XL-Sum	機械読解 JSQuAD	数学 MGSM	コード生成 JHumanEval	swallow-evaluation 日本語	swallow-evaluation 英語	教養科目 pfgen-bench
Llama 3 8B (ベース LLM)		0.836	0.445	0.400	0.456	0.220	0.209	0.176	0.888	0.332	**0.331**	0.429	**0.565**	0.403
分類器なし (ベースライン)		0.875	0.463	0.563	0.469	0.270	0.201	**0.212**	0.888	0.328	0.239	0.451	0.488	0.609
Wiki 分類器	Top 10%	0.891	**0.553**	**0.607**	0.484	**0.297**	**0.226**	0.209	0.895	0.284	0.241	**0.469**	0.499	0.639
	Top 10–30%	0.880	0.446	0.534	0.453	0.271	0.209	0.183	0.892	0.304	0.228	0.440	0.492	0.612
LLM 分類器	Top 10%	0.886	0.495	0.599	**0.502**	0.283	0.209	0.193	0.898	0.336	0.248	0.465	0.502	**0.665**
	Top 10–30%	**0.893**	0.501	0.562	0.460	0.281	0.205	0.191	**0.900**	**0.348**	0.257	0.460	0.496	0.626
Swallow コーパス v1 のヒューリスティックルールをそのまま適用する場合														
分類器なし		0.886	0.456	0.561	0.457	0.261	0.178	0.197	0.891	0.344	0.223	0.445	0.481	0.617
Wiki 分類器	Top 10%	0.885	0.534	0.595	0.464	0.280	0.185	0.197	0.892	0.272	0.231	0.453	0.482	0.640

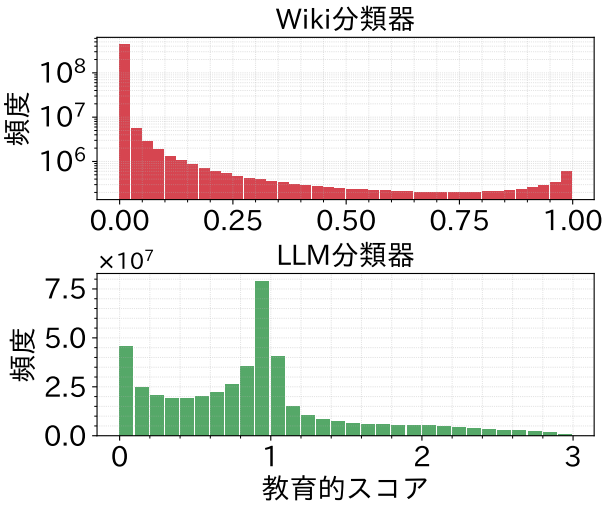

図 1　Swallow コーパス v2 の教育的スコアの分布

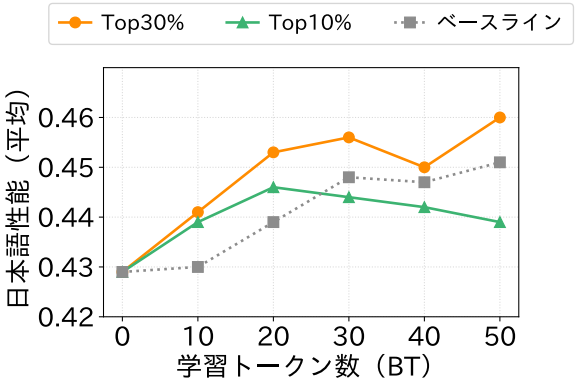

図 2　大規模学習を想定した模擬実験の日本語性能推移

4　実験

提案手法の有効性を検証するため，構築したコーパスを用いて LLM の継続事前学習を行った．ベース LLM には Llama 3 8B [13] を使用し，訓練データ量を 500 億トークン（50BT）とした．より実際のシナリオ[8]に近づけるため，学習時に日本語 Wikipedia（1.69BT）を混ぜることにし，Swallow コーパス v2 からは 48.31BT を取り出した．

評価には，llm-jp-eval [14] などの複数のツールを改変・統合した swallow-evaluation[9]（Ver. 202407）を用い，日英の幅広いタスクで LLM の性能を網羅的に評価した．評価タスクは Swallow の開発で共通に用いられている[10] 10 件の日本語理解・生成タスク [15, 16, 17, 18, 19, 20, 21, 22] と 9 件の英語理解・生成タスク [23, 24, 25, 26, 27, 28, 29, 30, 31] のほかに，今城らが作成した pfgen-bench [32] を追加で用いた．

8）Wikipedia テキストは Common Crawl ではなく Wikipedia のダンプから取り出す方が確実であるため，Swallow コーパスから除外している．
9）https://github.com/swallow-llm/swallow-evaluation
10）https://swallow-llm.github.io/evaluation/about.ja

4.1　品質フィルタリングの効果検証

提案した分類器やヒューリスティックルールを比較する実験を行った．分類器のスコアの閾値は上位 10%もしくは上位 10〜30%とした．また，2 節で述べたルールの効果を検証するため，Swallow コーパス v1 のルールを適用した場合も比較対象とした．

表 2 に各タスクの評価結果を示した（日本語タスクに関してはタスク毎のスコアも示した）．Wiki 分類器や LLM 分類器の上位 10%の文書を用いた場合，分類器を用いない場合と比較して日本語の知識に関するタスクを中心にスコアが改善しており，主に**教養科目**（JMMLU，pfgen-bench），**QA**（JCom.，JEMHopQA，NIILC），**英日翻訳**（WMT20）の伸びが大きい．特に JMMLU は 1.3 ポイントの伸びから 4.6 ポイントの伸びに改善しており，教育的な文書が専門知識の獲得に貢献することを示唆している．

一方で，要約（XL-Sum）や機械読解（JSQuAD）のような知識量よりも日本語の基礎能力が重要なタスクや，数学（MGSM）やコード生成（JHumanEval）のような言語個別性が低い推論能力を要求するタスク [33] に対して，提案手法の効果は限定的であった．4.3 節では，数学やソースコードの英語コーパスを併用した学習によって，日本語の数学やコード

表 3　Swallow モデル（8B）の日英ベンチマークスコア比較

モデル	分類器 Wiki	分類器 LLM	QA JCom.	QA JEMHopQA	QA NIILC	教養科目 JMMLU	英日翻訳 WMT20	日英翻訳 WMT20	要約 XL-Sum	機械読解 JSQuAD	数学 MGSM	コード生成 JHumanEval	swallow-evaluation 日本語	swallow-evaluation 英語	教養科目 pfgen-bench
Llama 3 (ベース LLM)	–	–	0.836	0.445	0.400	0.456	0.220	0.209	0.176	0.888	0.332	0.331	0.429	**0.565**	0.403
Llama 3 Swallow	×	×	0.895	0.485	0.564	0.470	0.276	0.222	0.198	0.895	0.424	0.289	0.472	0.542	0.646
Llama 3.1 (ベース LLM)	–	–	0.844	0.446	0.405	0.477	0.221	0.208	0.179	0.896	0.356	0.327	0.436	0.564	0.409
Llama 3.1 Swallow v0.1	○	×	**0.912**	0.509	0.601	0.518	0.291	0.231	**0.202**	**0.899**	0.460	0.281	0.491	0.558	0.671
Llama 3.1 Swallow v0.2	○	○	0.911	**0.510**	**0.627**	**0.525**	**0.296**	**0.233**	0.198	0.892	**0.464**	**0.336**	**0.499**	0.555	**0.704**

生成のタスク性能を補完できることを確認する．

Wiki 分類器と LLM 分類器の比較　2 つの分類器をスコア上位 10%で用いた際，平均スコアは同水準となるが，Wiki 分類器は翻訳タスク，LLM 分類器は教養科目タスクでの改善幅が大きい．これに対して，スコア上位 10〜30%を用いた場合，Wiki 分類器はベースラインを下回る結果であったが，LLM 分類器は上位 10%での改善傾向を維持し，ベースラインの性能を上回った．3.3 節で確認した通り，Wiki 分類器は Wikipedia と類似した文書の検出を想定しているため，教育的と見なす文書の範囲が狭い可能性がある．これに対し，LLM 分類器はより広範な文書で訓練されているため，この差が生じたと考えられる．ゆえに，LLM 分類器を主に採用し，Wiki 分類器を補助的に併用することが最良慣行であろう．

ヒューリスティックルールの調整　Swallow コーパス v1 のルールを採用した場合，分類器の適用有無によらず，教養科目（JMMLU）や日英翻訳（WMT20）などのタスクのスコアが下落した．この結果は，Swallow コーパス v1 のルールが適切ではないため，学習に有用な文書や対訳を含む文書を除去してしまったことを示唆している．ルールの調整が進み，より教育的な文書を厳選できるようになったことは，本研究の貢献の一つである．

4.2　大規模学習を想定した模擬実験

提案手法では教育的スコアの閾値を設定し，教育的と判断された文書のみを用いるため，学習データの品質は高まるが，量は減少してしまう．LLM の学習により多くの学習データが必要な場合，教育的スコアの閾値を維持したまま複数エポックの学習を行うのか，教育的スコアの閾値を下げてデータ量を増やすのか，選択を迫られる．そこで，Swallow コーパス v2 の全量相当（709BT）を，LLM 分類器でスコア上位 10%（93.1BT）または上位 30%（238.6BT）の文書のみで学習すると仮定し，必要なエポック数 T をそれぞれ算出する．そして，これらのエポック数 T で学習を行った場合にも提案手法が有効性を保てるかどうか検証した．ただし，709BT の学習はコストが高すぎるので，T エポックの学習で合計 48.31BT の学習量となるようにデータをサブセット化し（付録 C.2 参照），4.1 節と同様の実験を行った．

学習トークン数を変化させたときの swallow-evaluation の日本語平均スコアを図 2 に示す．LLM 分類器のスコア上位 30%を用いた学習では，依然としてベースラインを上回る結果が得られ，学習トークン数を増やす場合においても提案手法が有効である．一方で，スコア上位 10%のみを用いた場合は，エポック数が 3.15〜7.62 に相当する 20BT 以降でスコアが明確な下落傾向に転じ，最終的にベースラインを下回った．これらの結果や Muennighoff ら [34] の報告を参考にすると，エポック数が 4 を超えるようなアップサンプリングは避けるべきであろう．

4.3　Llama 3.1 Swalow の構築

HuggingFace 上で公開されている Llama 3.1 Swallow には本研究の成果が取り込まれている．v0.1[11] では Wiki 分類器のみ，v0.2[12] では両方の分類器が使われている．v0.1 と v0.2 の学習設定は他にも異なる点（付録 C.3）があるため，単純な比較はできない．ただ，表 3 に示すように，提案手法を適用したモデルは QA，教養科目，翻訳タスクのスコアの改善度合いが高まっている．また，Cosmopedia [35] や The Stack v2 [36] など英語コーパスの併用により，いずれのモデルも数学やコード生成，英語全般のスコアの劣化を防ぐか，改善を達成している．

5　おわりに

本研究では，文書の教育的価値に着目した分類器を品質フィルタリングに用い，Swallow コーパス v2 を構築した．提案手法はシンプルながら LLM の日本語能力の向上を達成し，複数の実験を通じて有効性を確認した．今後は，本研究で抽出した教育的な文書と LLM を用い，事前学習データの自動合成 [37] などでコーパスの品質と量を改善していきたい．

11）https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-8B-v0.1
12）https://huggingface.co/tokyotech-llm/Llama-3.1-Swallow-8B-v0.2

謝辞

この成果は, 国立研究開発法人新エネルギー・産業技術総合開発機構（NEDO）の委託業務（JPNP18002）の結果得られたものです．また，LLM の継続事前学習の実験では，国立研究開発法人産業技術総合研究所が構築・運用する AI 橋渡しクラウド（ABCI: AI Bridging Cloud Infrastructure）の「大規模言語モデル構築支援プログラム」の支援を受けました．この成果は，産総研政策予算プロジェクト「フィジカル領域の生成 AI 基盤モデルに関する研究開発」の結果得られました．

参考文献

[1] Jared Kaplan, Sam McCandlish, Tom Henighan, et al. Scaling laws for neural language models. arXiv:2001.08361, 2020.

[2] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, et al. The RefinedWeb dataset for Falcon LLM: Outperforming curated corpora with web data, and web data only. arXiv:2306.01116, 2023.

[3] Maurice Weber, Daniel Y. Fu, Quentin Anthony, et al. RedPajama: an open dataset for training large language models. In **NeurIPS Datasets and Benchmarks Track**, 2024.

[4] Naoaki Okazaki, Kakeru Hattori, Hirai Shota, et al. Building a large Japanese Web corpus for large language models. arXiv:2404.17733, 2024.

[5] 新里顕大, 高瀬翔, 清野舜ほか. 日本語 LLM 構築におけるコーパスクリーニングの網羅的評価. 言語処理学会第 30 回年次大会 (NLP2024), 2024.

[6] 榎本倫太郎, Tolmachev Arseny, 新妻巧朗ほか. 大規模言語モデル開発における日本語 Web 文書のフィルタリング手法の検証. 言語処理学会第 30 回年次大会 (NLP2024), 2024.

[7] Arseny Tolmachev, Masayoshi Hayashi, Takuro Niitsuma, et al. Uzushio: A distributed huge corpus processor for the LLM era. 言語処理学会第 30 回年次大会 (NLP2024), 2024.

[8] Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, et al. The FineWeb datasets: Decanting the Web for the finest text data at scale. arXiv:2406.17557, 2024.

[9] Jeffrey Li, Alex Fang, Georgios Smyrnis, et al. DataComp-LM: In search of the next generation of training sets for language models. arXiv:2406.11794, 2024.

[10] Teknium. OpenHermes 2.5: An open dataset of synthetic data for generalist LLM assistants. HuggingFace, 2023.

[11] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. arXiv:1607.04606, 2017.

[12] Luca Soldaini, Rodney Kinney, Akshita Bhagia, et al. Dolma: an open corpus of three trillion tokens for language model pretraining research. arXiv:2402.00159, 2024.

[13] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. arXiv:2407.21783, 2024.

[14] Namgi Han, 植田暢大, 大嶽匡俊ほか. llm-jp-eval: 日本語大規模言語モデルの自動評価ツール. 言語処理学会第 30 回年次大会 (NLP2024), 2024.

[15] Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. JGLUE: Japanese general language understanding evaluation. In **Proc. of LREC**, 2022.

[16] 石井愛, 井之上直也, 関根聡. 根拠を説明可能な質問応答システムのための日本語マルチホップ QA データセット構築. 言語処理学会第 29 回年次大会 (NLP2023), 2023.

[17] 関根聡. 百科事典を対象とした質問応答システムの開発. 言語処理学会第 9 回年次大会 (NLP2003), 2003.

[18] Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, et al. XL-Sum: Large-scale multilingual abstractive summarization for 44 languages. In **Findings of ACL-IJCNLP 2021**, 2021.

[19] Freda Shi, Mirac Suzgun, Markus Freitag, et al. Language models are multilingual chain-of-thought reasoners. In **ICLR**, 2023.

[20] Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, et al. Findings of the 2020 conference on machine translation (WMT20). In **The Fifth Conference on Machine Translation**, Online, 2020. Association for Computational Linguistics.

[21] 佐藤美唯, 志歩, 梶浦照乃, 倉光君郎. LLM は日本語追加学習により言語間知識転移を起こすのか？言語処理学会第 30 回年次大会 (NLP2024), 2024.

[22] 尹子旗, 王昊, 堀尾海斗, 河原大輔, 関根聡. プロンプトの丁寧さと大規模言語モデルの性能の関係検証. 言語処理学会第 30 回年次大会 (NLP2024), 2024.

[23] Todor Mihaylov, Peter Clark, Tushar Khot, et al. Can a suit of armor conduct electricity? a new dataset for open book question answering. In **The 2018 Conference on Empirical Methods in Natural Language Processing**, Brussels, Belgium, 2018. Association for Computational Linguistics.

[24] Mandar Joshi, Eunsol Choi, Daniel Weld, et al. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In **The 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, Vancouver, Canada, 2017. Association for Computational Linguistics.

[25] Rowan Zellers, Ari Holtzman, Yonatan Bisk, et al. HellaSwag: Can a machine really finish your sentence? In **The 57th Annual Meeting of the Association for Computational Linguistics**. Association for Computational Linguistics, July 2019.

[26] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In **The 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018**. Association for Computational Linguistics, 2018.

[27] Alexey Tikhonov and Max Ryabinin. It's all in the heads: Using attention heads as a baseline for cross-lingual transfer in commonsense reasoning. In **Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021**, Vol. ACL/IJCNLP 2021 of **Findings of ACL**. Association for Computational Linguistics, 2021.

[28] Dan Hendrycks, Collin Burns, Steven Basart, et al. Measuring massive multitask language understanding. arXiv:2009.03300, 2021.

[29] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. Training verifiers to solve math word problems. arXiv:2110.14168, 2021.

[30] Mirac Suzgun, Nathan Scales, Nathanael Schärli, et al. Challenging BIG-Bench tasks and whether chain-of-thought can solve them. In **Findings of the Association for Computational Linguistics: ACL 2023**. Association for Computational Linguistics, July 2023.

[31] Mark Chen, Jerry Tworek, Heewoo Jun, et al. Evaluating large language models trained on code. arXiv:2107.03374, 2021.

[32] 今城健太郎, 平野正徳, 鈴木脩司, 三上裕明. 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, 2024.

[33] Koshiro Saito, Sakae Mizuki, Masanari Ohi, et al. Why we build local large language models: An observational analysis from 35 Japanese and multilingual LLMs. arXiv:2412.14471, 2024.

[34] Niklas Muennighoff, Alexander M. Rush, Boaz Barak, et al. Scaling data-constrained language models. arXiv:2305.16264, 2023.

[35] Loubna Ben Allal, Anton Lozhkov, Penedo, et al. Cosmopedia. HuggingFace, 2024.

[36] Anton Lozhkov, Raymond Li, Loubna Ben Allal, et al. StarCoder 2 and The Stack v2: The next generation. arXiv:2402.19173, 2024.

[37] Marah Abdin, Jyoti Aneja, Harkirat Behl, et al. Phi-4 technical report. arXiv:2412.08905, 2024.

A　ヒューリスティックルール

Swallow コーパス v1 および v2 で採用されているヒューリスティックルールの一覧を表 4 に示す．v2 では v1 のルールの一部を廃止し，有用な文書が過剰に除去されないようにした．

表 4　ヒューリスティック・ルールの一覧

ルール	閾値 v1	閾値 v2
文や文字の品質に関するルール		
日本語の文字数	400 未満	
ひらがな文字の割合	0.20 未満	
カタカナ文字の割合	0.50 以上	廃止
日本語文字の割合	0.50 未満	
文の平均文字数	90 未満／200 以上	
最も長い文の文字数	200 以上	廃止
省略記号で終了する文の割合	0.20 以上	
単語や行の重複に関するルール		
他と重複する行／段落の割合	0.30 以上	
他と重複する行／段落に含まれる文字割合	0.20 以上	
最頻出の 2-gram の出現回数割合	0.20 以上	
最頻出の 3-gram の出現回数割合	0.18 以上	
最頻出の 4-gram の出現回数割合	0.16 以上	
複数回出現する 5-gram の出現回数割合	0.15 以上	廃止
複数回出現する 6-gram の出現回数割合	0.14 以上	
複数回出現する 7-gram の出現回数割合	0.13 以上	
複数回出現する 8-gram の出現回数割合	0.12 以上	
複数回出現する 9-gram の出現回数割合	0.11 以上	
複数回出現する 10-gram の出現回数割合	0.10 以上	
有害性に関するルール		
NG 表現の文字の割合	0.05 以上	

B　LLM 分類器の詳細

Llama 3.1 70B Instruct で LLM 分類器の訓練データに教育的スコアを付与させた際のプロンプトを図 3 に示す．また，実際に付与された教育的スコアの分布を表 5 に示す．

```
Below is an extract from a web page. As an experienced teacher with a focus on higher education, evaluate the educational value of the given text using the additive 3-point scoring system described below.

### Evaluation Criteria

1. Highly Educational Topic (1 point):
   - The extract provides objective facts or knowledge that are important for university students to acquire a broad education and has high educational value. It helps build a crucial foundation for academic learning and social life and has broad applicability. For example, it includes knowledge related to business, accounting, philosophy, everyday trivia, science, social sciences, humanities, law, technology, health, etc.
2. Provides Deep Insights or Discussions (1 point):
   - The extract consistently offers detailed information and explanations on educational topics. It goes beyond merely handling words or concepts superficially, providing deep insights or discussions, and thus has high educational value.
3. Clear Explanation for General Audience (1 point):
   - The extract provides clear and simple explanations on educational topics, making it easy for the general public, who are not experts in the field, to understand the content well.

### Evaluation Method

1. Evaluate the text on a 3-point scale based on the above criteria.
2. Add 1 point for each criterion met (a maximum of 3 points if all criteria are met).

### Output Format

1. First, briefly explain the evaluation results (0 points or 1 point) for each of the three criteria and the reasons for each.
2. Finally, state the total score in the format "Educational Score: <total points>".

### Extract

{TEXT}

### Output
```

図 3　教育的スコアの付与に用いたプロンプト

表 5　LLM 分類器の訓練データの教育的スコア分布

データ	教育的スコア 0	1	2	3
Swallow コーパス v2	62,768	104,215	20,916	12,081
独自選定したウェブ記事	1,437	4,961	14,684	9,977

C　実験で用いたコーパス

C.1　Swallow コーパス v2 の規模

2 節で抽出し，4.1 節で品質フィルタリングを適用した Swallow コーパス v2 の規模を表 6 に示す．

表 6　Swallow コーパス v2 の規模

フィルタリング 分類器		ルール	分量 トークン	分量 ページ
分類器なし	–	なし	1,695.98BT	1,907,810,987
		厳格（v1）	467.18BT	359,459,885
		通常（v2）	709.22BT	454,928,974
Wiki 分類器	Top10%	厳格（v1）	59.94BT	34,035,606
	Top10%	通常（v2）	63.28BT	43,429,972
	Top10–30%	通常（v2）	147.11BT	89,904,750
LLM 分類器	Top10%	通常（v2）	93.10BT	45,054,110
	Top10–30%	通常（v2）	145.51BT	90,146,778

C.2　大規模学習を想定した模擬実験

章 4.2 の実験に用いたコーパスの配合を表 7 に示す．例えば，LLM 分類器スコア上位 10%の文書（93.1BT）のみで Swallow コーパス v2 全量相当（709BT）を学習するのに必要なエポック数 T は 7.62 であるため，実験でもこのエポック数 T を反映できるようにサブセットの量を 6.34BT とした（6.34BT×7.62=48.31BT）．

表 7　大規模学習を想定した模擬実験のコーパス配合

実験設定	Top10%のサブセット トークン	エポック	比率	Top10-30%のサブセット トークン	エポック	比率
Top30%	6.34BT	2.54	33%	9.91BT	3.25	67%
Top10%	6.34BT	7.62	100%	-	-	-

C.3　Swallow モデルの学習

4.3 節で説明した Llama 3/3.1 Swallow の学習に用いたコーパスの配合を表 8 に示す．実際にはこれらのコーパスを用いて，Llama 3 Swallow は 100BT，Llama 3.1 Swallow v0.1/v0.2 は 230BT，252BT まで学習を行った．よって，各モデルに用いた日本語テキストの量は単純計算で 82.5BT，124.2BT，172.7BT に相当する．

表 8　Swallow モデルのコーパス配合

コーパス	トークン（BT） 3	3.1 v0.1	3.1 v0.2
日本語	**82.5**	**108.0**	**137.1**
Swallow コーパス v1	80.8	–	–
Swallow コーパス v2			
Wiki 分類器 Top10%	–	63.3	40.6
LLM 分類器 Top10%	–	–	93.1
分類器なし	–	41.3	–
日本語 Wikipedia	1.7	3.4	3.4
英語	**13.8**	**45.0**	**30.9**
ソースコード	**3.7**	**32.0**	**32.0**
合計	**100.0**	**200.0**	**200.0**