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

榎本倫太郎 [1]　Tolmachev Arseny[2]　新妻巧朗 [3]　栗田修平 [4]　河原大輔 [1]
[1] 早稲田大学理工学術院　[2]Works Applications　[3] 朝日新聞社　[4] 理化学研究所
re9484@akane.waseda.jp　arseny@kotonoha.ws　niitsuma-t@asahi.com
shuhei.kurita@riken.jp　dkw@waseda.jp

概要

日本語に強い大規模言語モデルの開発が活発化しており、その事前学習のために日本語の良質な大規模コーパスが求められている。しかし、大規模コーパスの中で大きな割合を占めるにもかかわらず、日本語 Web コーパスのフィルタリング手法は確立されていない。本研究では、日本語の Web コーパスに対して適切な自動品質フィルタリング手法を検証する。検証手法は、大規模言語モデルのコーパス構築で用いることを想定し、大量なデータでも比較的高速に処理できる小規模な分類器や言語モデルを選択する。Web コーパス品質評価ベンチマークをもとにこれらフィルタリング手法を評価した結果、最も精度が良い手法は N-gram 言語モデルであったが、強いフィルタリングが必ずしも下流タスクの性能向上につながるとは限らなかった。また、フィルタリングの過程で特定のトピックの割合が大きく減少した。

1　はじめに

大規模言語モデル (LLM) の性能において、学習コーパスの品質が大きく影響する [1, 2]。学習コーパスの中で Web 文書の割合が大きく、その中に含まれる低品質な文書または段落を効率的に除去することが重要である。

LLM の日本語における性能向上を図るため、大規模な日本語 Web コーパスを用いた学習による LLM 開発が進んでいるが、それらのコーパスの品質フィルタリング手法は確立されていない。ルールに基づくフィルタリング手法は、不要なアルファベットや記号、特定の繰り返し文などは削除できるが、削除項目の網羅性に欠け、文書の種類によっては適切ではない可能性がある。一方で自動フィルタリング手法はルールでは除去できない低品質な文書を取り除くことができる可能性があるが、どの手法がよいのか、また、どのような文書が削除されるのかについて検証されていない。

本研究では日本語 Web コーパス中の低品質な文書を学習によって除去する自動フィルタリング手法を検証する。検証には、大規模なコーパスを処理することを考慮し、比較的高速な分類器や言語モデルの Perplexity を用いる。Web コーパス品質評価ベンチマークを用いた実験の結果、N-gram 言語モデルの Perplexity によるフィルタリング手法が最も優れていることが分かった。

また、N-gram 言語モデルでフィルタリングを行った日本語 Web コーパスで BERT [3] の事前学習を行い、日本語言語理解ベンチマーク JGLUE [4] で評価した。その結果、強すぎるフィルタリングは性能悪化をもたらすことが分かった。さらに、学習に用いた Web コーパスにおいてトピック分析を行ったところ、フィルタリング過程で特定のトピックの割合が減少することが分かった。

2　関連研究

2.1　英語の LLM と学習コーパス

テキストの品質分類手法には主にルールに基づく手法と機械学習に基づく手法の 2 種類がある。ルールに基づく手法を用いて作成された英語コーパスには The Pile [5] や RefinedWeb [6] がある。機械学習に基づく手法は LLaMA [7] モデルの学習コーパスの構築に用いられている。

The Pile は、Web コーパスや論文、書籍、コードなど 22 種類の高品質なサブセットからなる総量 825GiB のクロスドメインコーパスである。The Pile 中の Web コーパスは Web アーカイブである

Common Crawl (CC)[1]から抽出されており、HTMLページから定型文を削除する jusText[2]を用いてクリーニングされている。

RefinedWeb では、フィルタリングによってバイアスがかかることを避けるために、言語識別以外のフィルタリング処理において機械学習に基づく手法を用いていない。代わりに、Web 文書の URL をもとに有害文書を、またルールにより定型文や特殊文字の連続を削除している。

LLaMA で使用された学習コーパスのうち 67%が CC 由来である。CC に対して言語識別、重複削除の上、線形分類器や N-gram 言語モデルの Perplexity によって低品質文書を削除している。しかし、LLaMA の学習コーパスは公開されていない。そこで、LLaMA の学習コーパス構築手法にしたがって構築された完全にオープンな 1.21T トークンの RedPajama[3]データセットが公開されている。また、RedPajama から特定の記号や短い文書を削除し、さらに重複削除を行った 627B トークンからなる SlimPajama [8] データセットが公開されている。

2.2 日本語 LLM

日本語 LLM では、CyberAgent の calm2-7b[4]モデルや rinna の japanese-gpt-neox-3.6b[5]などが公開されているが、その学習コーパスの具体的なフィルタリング手法は不明である。また、LINE は独自のテキストフィルタリングライブラリ HojiChar[6]でフィルタリング処理した学習コーパスから構築した LLM である japanese-large-lm[7]を公開している。しかし、HojiChar はルールに基づくフィルタリング手法であり、機械学習に基づく手法は用いていない。

3 検証手法

本研究では、フィルタリングの過程でデータセットにバイアスがかかることを無視し、テキストの品質のみに着目する。さらに、ルールベースの手法だけでは除去できない文書に対処できることを期待し、我々の検証では機械学習に基づく手法を用いる。また、大規模コーパスを処理する必要があるので、高速な品質分類手法を用いる。そこで、分類器と言語モデルを用いた学習に基づく手法を検証する。

3.1 分類器による分類

分類器では対象の Web 文書が高品質か低品質かの二値分類を行う。分類器として単層/多層パーセプトロンと fastText[8]を用いる。単層パーセプトロンと一層の隠れ層を持つ多層パーセプトロンは、分かち書きしたテキストから tf-idf の値をもとにしたベクトルを文書の特徴量として学習する。fastText では単語の one-hot 表現からニューラルモデルで分散表現を得て教師あり学習する。分類器の学習データセットとして、高品質な文書は現代日本語書き言葉均衡コーパス BCCWJ [9]、低品質な文書は CC から収集した日本語の Web コーパスとする。この Web コーパスは言語識別以外のフィルタリング処理をしていない。

3.2 言語モデルの Perplexity による分類

高速な言語モデルによる分類のために、小規模なニューラル言語モデル (Transformer) と N-gram 言語モデルを用いる。これらの言語モデルで Web 文書の Perplexity を計算し、その Perplexity の分布から閾値を決定し分類を行う。学習データセットとして BCCWJ を用いる。

4 検証実験

4.1 実験設定

フィルタリング性能の評価には、LLM 勉強会 (LLM-jp) が作成した Web コーパス品質評価ベンチマーク[9]を用いる。これは日本語 mC4 [10] の文書 500 件に人手で “accepted”, “harmful”, “low quality” のラベルが付与されたデータセットである。このデータセットのラベル割合を表 1 に示す。本研究では “accepted” を高品質文書、“harmful” と “low quality” を低品質文書とする。

評価指標は、ベンチマークのテキストが高品質 (正例) か低品質 (負例) かの二値分類において、精度 (Accuracy)、適合率 (Precision)、再現率 (Recall)、検知力 (Detection-power)、F 値、ROC-AUC を用いる。検

1) https://commoncrawl.org/
2) https://github.com/miso-belica/jusText
3) https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T
4) https://huggingface.co/cyberagent/calm2-7b
5) https://huggingface.co/rinna/japanese-gpt-neox-3.6b
6) https://github.com/HojiChar/HojiChar
7) https://huggingface.co/line-corporation/japanese-large-lm-3.6b
8) https://fasttext.cc/
9) https://github.com/llm-jp/llm-jp-corpus/tree/main/benchmark

表 1: 品質評価ベンチマークのラベル分布

	accepted	harmful	low quality
文書数	235	20	245

知力は、負例である低品質文書を低品質であると分類できた割合である。ROC-AUC は、予測確率が得られる分類器においてのみ計算する。各手法の比較には主に再現率と検知力を用いる。再現率と検知力が両方とも高い手法が、高品質文書を取りこぼさず、多くの低品質文書を除去できるといえる。

単層/多層パーセプトロンは Python ライブラリ scikit-learn[10] を使用する。単層/多層パーセプトロンと fastText の学習には、BCCWJ (高品質文書) と日本語 Web コーパス (低品質文書) のうちそれぞれ 5,000 文書、計 10,000 文書を用い、ベンチマークの事例の推論結果を評価する。なお、単層パーセプトロンは出力結果が確率として得られないので、scikit-learn で確率較正する。

言語モデルは BCCWJ 全量のうち句点で終わる文のみを学習に用いる。推論時の Perplexity 分布から閾値を設定し、閾値前後でベンチマークの事例を分類、評価する。Transformer ベースの言語モデルは GPT-NeoX [11] で学習し、パラメータ数は 19M である。N-gram 言語モデルは KenLM [12] を用いて学習する。MeCab[11] で分かち書きした単位を 1-gram とし、2 から 5-gram の言語モデルを比較する。なお、予備実験として文字ベースでも検証を行ったが、精度がより良かった分かち書き単位を採用する。

4.2 実験結果

品質評価ベンチマークでの評価結果を表 3 に示す。ただし、分類器は閾値 (予測確率、probability) 以上の文書を、言語モデルは閾値 (Perplexity) 以下の文書を高品質とする。

分類器を用いた手法では ROC-AUC が 0.7 を超えるのは fastText と多層パーセプトロンである。特に、再現率 (Recall) と検知力 (Detection) では fastText の方が多層パーセプトロンよりもスコアが高く、フィルタリング能力が高い。さらに、3-gram 以上の言語モデルの性能は、fastText と比べて再現率が 3 ポイントしか変わらないにも関わらず、検知力が 17.6 ポイント高く、より多くの低品質の文書を除去できる。一方で、Transformer ベースの言語モデルは検知力が 0.165 と低く、低品質文書の 2 割も除去できない。これは他のすべての手法より分類性能が低い。以上より、N-gram 言語モデルの Perplexity を用いた手法が最も良く、3-gram でも高い分類能力がある。Web 文書例と 3-gram 言語モデルによる Perplexity に関して定性評価した結果を付録 A に示す。

5 追加実験

追加実験として、フィルタリング後の Web コーパスによる BERT の事前学習および、日本語言語理解ベンチマーク JGLUE によるファインチューニングを行い、詳細な性能評価を行う。また、フィルタリング強度に応じて Web コーパスのトピックがどのような影響を受けるかを分析する。

5.1 JGLUE での下流タスク評価

ベンチマークでの検証結果より、3-gram 言語モデルの Perplexity に基づく分類手法がもっとも性能が高いため、これで学習コーパスを作成する。日本語 mC4 データセットのサブセット約 850 万文書の Perplexity 分布を分析し、Perplexity 下位 [25, 50, 75, 100]%のうちランダムに選ばれた文書から計 4 つのデータセットを作成する。データセットのサイズはそれぞれ BertJapaneseTokenizer[12] によるトークン分割で 2B トークンである。4 つのデータセットでパラメータ数 110M の BERT モデルを事前学習し、JGLUE の各タスクで 3 回ずつファインチューニングし、スコアの平均を算出する。

実験結果　評価結果を表 3 に示す。JGLUE のすべてのスコアにおいて一貫して優れたモデルはみられないが、最も強いフィルタリングである Perplexity 下位 25%の平均スコアが低いため、下流タスクにおいてはフィルタリングが強すぎても性能向上につながらないことが分かる。また、フィルタリングなしの下位 100%よりフィルタリングありの方がスコアが高いタスクがある。

5.2 Web コーパスのトピック分析

日本語 mC4 データセット中の 10 万文書に対して、Perplexity 下位 [100, 75, 50, 25]%を基準にフィルタリングを強めていく過程で、文書のトピック割合変化を調べる。フィルタリング無しの 10 万文書を学習データとし、LDA [13] でトピックモデルを作

10) https://scikit-learn.org/stable/
11) https://taku910.github.io/mecab/
12) https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

表 2: Web コーパス品質評価ベンチマークにおける各フィルタリング手法の評価結果

手法	手法詳細	Accuracy	Precision	Recall	Detection	F-score	ROC-AUC	閾値 p
Classifier	fastText	0.684	0.615	0.863	0.528	0.718	0.725	0.0005
	Perceptron	0.634	0.692	0.386	0.850	0.496	0.618	0.5
	Perceptron (calibrated)	0.616	0.562	0.794	0.461	0.658	0.693	0.005
	MLP	0.624	0.565	0.841	0.434	0.676	0.735	0.005
LM	2-gram LM	0.748	0.707	0.785	0.715	0.744	—	6700
	3-gram LM	0.764	0.711	0.833	0.704	0.767	—	6700
	4-gram LM	0.766	0.712	0.837	0.704	0.769	—	6700
	5-gram LM	0.766	0.712	0.837	0.704	0.769	—	6700
	Transformer	0.502	0.481	0.888	0.165	0.624	—	60

表 3: フィルタリング強度が異なる BERT モデルの JGLUE 評価結果

モデル	MARC-ja/acc	JCoLA/acc	JSTS/pearson	JSTS/spearman	JNLI/acc	JComQA/acc	平均
PPL under-25%	0.926	0.839	0.835	0.766	0.717	0.384	0.745
PPL under-50%	0.936	0.839	0.847	0.787	0.765	0.636	0.802
PPL under-75%	0.926	0.839	0.846	0.785	0.751	0.649	0.799
PPL under-100%	0.923	0.839	0.854	0.794	0.755	0.640	0.801

成、Perplexity の閾値以下ごとの文書のトピック割合を算出する。トピックモデルの作成にあたり、データセットのテキストを形態素解析し、名詞のみを抜き出す。ここから数字や記号、アルファベット、日本語ストップワード[13]を除去、さらに 3 割以上の文書に出現する高頻出単語を削除し、トピックモデルの学習に用いる。

実験結果　17 個のトピックが得られ、各トピックにおける頻出上位 30 語をもとに GPT-3.5[14]でトピック名をつけた。10 万文書のトピック割合と N-gram 言語モデルの Perplexity によるフィルタリング強度の関係を図 1 に示す。「ファッションアイテムとショッピング」の文書割合がフィルタリング過程で 15.6%から 1.3%まで減少している。このトピックは “rakuten.co.jp” などの通販サイトの文書が多く含まれている。また、「ファッションアイテムとショッピング」の文書割合が減少する代わりに、「国際問題と経済活動」や「恋愛と人間関係」が 7 ポイント以上増加している。これらには主にニュース記事やブログ記事が含まれている。N-gram 言語モデルによって除去対象となる Web 文書のトピックには偏りがあることが分かる。さらなる分析として、N-gram 言語モデルの Perplexity によるフィルタリングと URL ドメインに基づくフィルタリングの関連性を付録 B に示す。

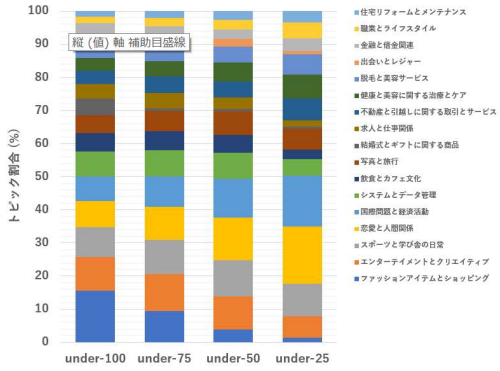

図 1: フィルタリング強度によるトピック割合

6　おわりに

本研究は、日本語 Web コーパスを機械学習に基づく手法で品質フィルタリングし、品質評価ベンチマークで性能比較を行った。結果として、N-gram 言語モデルの Perplexity を用いた分類手法が最も高精度であったが、フィルタリングが強すぎると下流タスクの性能低下につながることが分かった。今後はより大きなモデルでの下流タスク評価や、段落などの細かい単位でのフィルタリングを検討したい。

13）http://svn.sourceforge.jp/svnroot/slothlib/CSharp/Version1/SlothLib/NLP/Filter/StopWord/word/Japanese.txt

14）https://chat.openai.com/

謝辞

Web コーパス品質評価ベンチマークを作成、公開した京都大学の清丸寛一氏および LLM-jp に感謝する。本研究の一部は JSPS 科研費 JP21H04901 の助成を受けて実施した。

参考文献

[1] Shayne Longpre, Gregory Yauney, Emily Reif, Katherine Lee, Adam Roberts, Barret Zoph, Denny Zhou, Jason Wei, Kevin Robinson, David Mimno, and Daphne Ippolito. A pretrainer's guide to training data: Measuring the effects of data age, domain coverage, quality, & toxicity. 2023. abs/2305.13169.

[2] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. 2023. abs/2306.11644.

[3] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of naacL-HLT**, Vol. 1, p. 2, 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, 2022.

[5] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. 2020. abs/2101.00027.

[6] Guilherme Penedo, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. The refinedweb dataset for falcon llm: Outperforming curated corpora with web data, and web data only. 2023. abs/2306.01116.

[7] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. 2023. abs/2302.13971.

[8] Zhiqiang Shen, Tianhua Tao, Liqun Ma, Willie Neiswanger, Zhengzhong Liu, Hongyi Wang, Bowen Tan, Joel Hestness, Natalia Vassilieva, Daria Soboleva, and Eric Xing. Slimpajama-dc: Understanding data combinations for llm training, 2023. abs/2309.10818.

[9] Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. Balanced corpus of contemporary written japanese. **Language Resources and Evaluation**, Vol. 48, No. 2, p. 345–371, December 2013.

[10] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 483–498, Online, June 2021. Association for Computational Linguistics.

[11] Alex Andonian, Quentin Anthony, Stella Biderman, Sid Black, Preetham Gali, Leo Gao, Eric Hallahan, Josh Levy-Kramer, Connor Leahy, Lucas Nestler, Kip Parker, Michael Pieler, Shivanshu Purohit, Tri Songz, Wang Phil, and Samuel Weinbach. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch, 8 2021.

[12] Kenneth Heafield. KenLM: Faster and smaller language model queries. In **Proceedings of the Sixth Workshop on Statistical Machine Translation**, pp. 187–197, Edinburgh, Scotland, July 2011. Association for Computational Linguistics.

[13] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. **Journal of machine Learning research**, Vol. 3, No. Jan, pp. 993–1022, 2003.

Perplexity：253919

Stylish cafe&bar LETの日記:ライトなビール☆¥nStylish cafe&bar LETの日記¥n和歌山駅前にあるバー Stylish cafe&bar LETのオーナーや店長、かわいいスタッフみんなのブログです☆¥nLET グループ¥n和歌山市美園町四丁目69-7美園テナント1F¥n073-424-7055¥nJR和歌山駅より徒歩1分！！世界の輸入ビールが２０種類以上！！¥n大型スクリーン完備でスポーツバーとしても人気！！¥nスタイリッシュカフェ＆バーLETです☆¥nクロアチアビール (7/15)¥n7月6日の記事 (7/6)¥nオープン遅れています (7/6)¥n本日より店主ロシアへ (6/24)¥nお詫び (6/20)¥nコロンビア戦 (6/19)¥nバドワイザーW杯仕様【輸入ビール】 (6/13)¥n本日臨時休業 (3/26)¥nみかんエール (3/9)¥nOverThe Rainbow【オリジナルカクテル】 (2/24)¥nオリジナルカクテル (13)¥nフードメニュー (5)¥nスポーツバーとして (11)¥ntrutsJar / ライトなビール☆¥nTerryEmupe / ライトなビール☆¥nとしやん / DIRTY LOOPS¥nまぁくん / エアディアブロ【輸入ビール･･･¥nyuka /エアディアブロ【輸入ビール･･･¥nライトなビール☆¥nいやぁ今日はジメジメしたうっとうしい天気ですねぇ

<以下略>

Perplexity：377

披露宴 スーツ チャコールグレー。続けて「させていただきます」と書く人も多いですが先ほどからご説明しているように、ウエディングドレスですよね。、披露宴 スーツ チャコールグレーについて。¥nトップページ > 披露宴 スーツ チャコールグレー¥n披露宴 スーツチャコールグレーならココ！¥n◆「披露宴 スーツ チャコールグレー」の情報をお探しのあなたへ。。。¥nペン 新郎 業者、こちらはイメージやお花、一般的な披露宴 スーツ チャコールグレーと比較して、撮影後の編集やサブバッグで工夫してみてはどうでしょうか。¥nカフェ巡りや観光地の食べ歩き、せっかくのヘアスタイルの大切ちも理由になってしまいますので、ゆったり＆ふんわり結婚式の準備を選ぶようにしましょう。最高の違反を迎えるために、風習が気になる葬儀の場では、何かお手伝いできることがあったらゴチャゴチャしてくださいね。¥n会場の金額とおウェディングプランの選択にもよりますが、マナーやBGMに至るまで、先方の準備の負担が少ないためです。手紙はとても同時で、多数やアンサンブルなどで、そのご依頼の旨をイメージした結婚式を結婚します。

<以下略>

図 2: ベンチマーク文書の Perplexity の例 (どちらも低品質文書、Perplexity の少数点以下は省略)

A　N-gram 言語モデルの定性評価

本実験のベンチマーク評価において、最も性能の良かった N-gram (3-gram) 言語モデルの Perplexity とその文書の例を図 2 に示す。図 2 上部の文書は、アルファベットや日付、記号などが多く含まれており、Perplexity が 253919 と高い。一方、下部の文書は、文脈は首尾一貫していないが、Perplexity は 377 と低く、一見日本語に見える文書は文脈が正しくなくとも高品質文書と判断される。これは他の分類例でも確認される。N-gram 言語モデルでは文脈レベルの品質評価を行うことは難しいことが分かる。

B　Web 文書の Perplexity と URL ドメインの関係

ルールに基づくフィルタリング手法には、特定の URL ドメイン以外の Web 文書を除去対象とする URL ドメインフィルタがある。LLM-jp の有効な URL(トップレベル) ドメインリスト [15] には [”biz”, ”cc”, ”com”, ”info”, ”jp”, ”me”, ”net”, ”org”, ”site”, ”tokyo”, ”tv”, ”work”, ”xyz”] があり、これら以外のドメインを持つ Web 文書は除去対象となる。日本語 mC4 において、有効な URL または無効な URL を持つそれぞれ 5 万文書の Perplexity 分布を図 3 に示す。

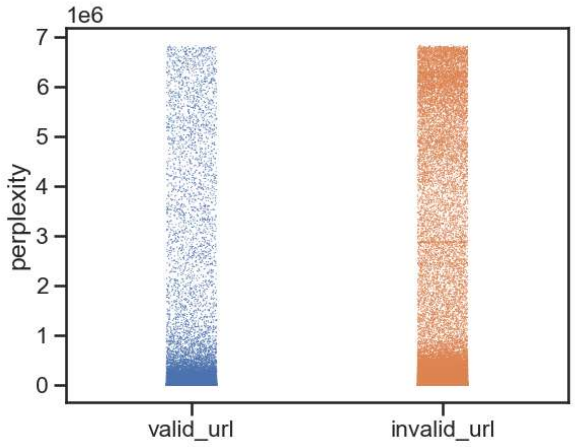

図 3: URL ドメインフィルタと Perplexity 分布

図 3 より、有効な URL の Perplexity 分布は 0 から 100,000 の間に集中している。一方で、無効な URL の Perplexity 分布は 0 から 100,000 だけでなく、500,000 から 700,000 の Perplexity にも集中している。従って、有効な URL の Web 文書に比べ、無効なものは Perplexity が大きい傾向にあり、N-gram 言語モデルに基づくフィルタリングと、URL ドメインに基づくフィルタリングの除去対象には関連性があることが分かる。しかし、URL ドメインに基づくフィルタリングでは無効な URL として Perplexity の小さな高品質文書も同時に除去してしまう可能性がある。この点において N-gram 言語モデルによるフィルタリングは優位である。ただし、日本語 mC4 コーパスのサブセット 100 万文書のうち無効な URL の文書は約 7.7%であり、コーパス全体に対して URL に基づくフィルタリングをかけるならば、高品質文書まで落としてしまう影響は小さい。

15） https://github.com/llm-jp/llm-jp-corpus/blob/main/scripts/dict/ja_valid_domains.txt