Investigating Web Corpus Filtering Methods for Language Model Development in Japanese

Rintaro Enomoto[1] **Tolmachev Arseny**[2*]
Takuto Niitsuma[3] **Shuhei Kurita**[4] **Daisuke Kawahara**[1,4,5]
[1]Waseda University [2]Works Applications
[3]The Asahi Shimbun Company [4]National Institute of Informatics
[5]LLMC, National Institute of Informatics
re9484@akane.waseda.jp arseny@kotonoha.ws
niitsuma-t@asahi.com skurita@nii.ac.jp dkw@waseda.jp

Abstract

The development of large language models (LLMs) is becoming increasingly significant, and there is a demand for high-quality, large-scale corpora for their pretraining. The quality of a web corpus is especially essential to improve the performance of LLMs because it accounts for a large proportion of the whole corpus. However, filtering methods for Web corpora have yet to be established. In this paper, we present empirical studies to reveal which filtering methods are indeed effective and analyze why they are. We build classifiers and language models in Japanese that can process large amounts of corpora rapidly enough for pretraining LLMs in limited computational resources. By evaluating these filtering methods based on a Web corpus quality evaluation benchmark, we reveal that the most accurate method is the N-gram language model. Indeed, we empirically present that strong filtering methods can rather lead to lesser performance in downstream tasks. We also report that the proportion of some specific topics in the processed documents decreases significantly during the filtering process.

1 Introduction

The quality of the pretraining data significantly impacts the performance of large language models (LLMs) (Longpre et al., 2023; Gunasekar et al., 2023). The training data mostly comprise Web documents, and therefore it is essential to remove low-quality documents or paragraphs from them efficiently. However, no quality filtering method has been established for these documents.

Rule-based filtering can quickly remove documents with many unnecessary alphabets, symbols, and specific repetitive sentences. However, they have no comprehensive understanding of documents to be removed and can overdo or overlook certain types of documents. On the other hand, learning-based filtering methods can remove some low-quality documents that seem to maintain a level of quality and therefore bypass the rule-based filters. However, it is not verified yet which filtering methods are better and what types of documents are removed by such filters.

In empirical experiments, We examine learning-based filtering methods to remove low-quality documents in a Web corpus. We focus on Japanese corpora in this paper because there has been little study on methods for filtering Japanese corpora for developing LLMs, while many models have been developed in recent years, especially for the Japanese language. We test the perplexity of a language model and a relatively fast classifier to process a massive volume of corpora. The experimental results show that the perplexity filtering method based on an N-gram language model is the best. We also pretrain BERT (Devlin et al., 2019) on a Japanese Web corpus filtered by the N-gram language model and evaluated it on Japanese General Language Understanding Evaluation, JGLUE (Kurihara et al., 2022). The results show that massively strong filtering results in performance deterioration. Furthermore, topic analysis on the Web corpus shows that the proportion of specific topics decreases during the filtering process.

2 Related Work

2.1 LLMs and Training Corpora in English

There are two main types of text quality classification methods: rule-based methods and learning-based methods. English corpora created using rule-based methods include the Pile (Gao et al., 2020) and RefinedWeb (Penedo et al., 2023). Learning-based methods are used to build training corpora for the LLaMA (Touvron et al., 2023) model.

The Pile is a cross-domain corpus with a total volume of 825 GiB, consisting of 22 high-quality

[*]Currently affiliated with Hakuhodo Technologies

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop), pages 154–160

subsets of web corpora, articles, books, and code. The Web corpus in the Pile is extracted from Common Crawl (CC)[1], which is a Web archive, and cleaned using jusText[2], which removes canned text from HTML pages.

RefinedWeb does not use learning-based methods in filtering processes except for language identification to avoid bias caused by filtering. Instead, harmful documents are removed based on a URL blacklist, and canned text and sequences of special characters are removed by rules.

Of the training corpus used for LLaMA, 67% is derived from CC. First, language identification and deduplication are applied to CC, and then low-quality documents are removed by linear classifiers and the perplexity of N-gram language models. However, the LLaMA training corpus is not publicly available. Instead, the fully open 1.21T-token RedPajama[3] dataset, built according to LLaMA's recipe, is publicly available. In addition, the SlimPajama (Shen et al., 2023) dataset, consisting of 627B tokens, has been published, which was constructed by removing certain symbols and short documents from RedPajama and further deleting duplicates.

2.2 Japanese LLMs

Japanese LLMs, such as CyberAgent's calm2-7b[4] and rinna's japanese-gpt-neox-3.6b[5], are publicly available, but the specific filtering method of their training corpus is unknown. LINE has also released japanese-large-lm[6], an LLM constructed from a training corpus filtered by its own text filtering library HojiChar[7]. It is a rule-based filtering method and does not use learning-based methods.

3 Investigation Method

In this study, we ignore any bias caused by a filtering process and focus only on the quality of a corpus. Furthermore, we use learning-based methods in our investigation in the hope that they can deal with documents that cannot be removed by rule-based methods alone. In addition, since we need to process a large corpus, we use a fast classification method. Therefore, we examine learning-based methods using a classifier and a language model.

3.1 Classification by Classifiers

Our classifier performs a binary classification of whether the target Web document is of high or low quality. The single / multilayer perceptron and fastText[8] are used as classifiers. The single-layer perceptron and the multilayer perceptron with one hidden layer are trained given a tf-idf vector extracted from a tokenized document. FastText is a supervised neural model based on a distributed representation obtained from a one-hot representation of words. For the training dataset for these classifiers, we prepared two types of data: high-quality and low-quality documents. We use Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa et al., 2013) for high-quality documents and a Japanese Web corpus collected from Common Crawl for low-quality documents. The latter documents are not filtered except for language identification.

3.2 Classification by Perplexity of Language Models

We use 6-layer 19M-parameter Transformer-based neural language models and N-gram language models. The perplexity of Web documents is calculated using these language models, and thresholds are determined from the distributions of the perplexity to perform classification. BCCWJ is used as the training dataset.

4 Experiments

4.1 Experimental Settings

To evaluate filtering performance, we use the Web Corpus Quality Evaluation Benchmark (WCQEB)[9], which was created by LLM-jp. This dataset consists of 500 Japanese mC4 (Xue et al., 2021) documents, each of which is manually labeled "accepted", "harmful", or "low quality". Table 1 shows the label distribution of this dataset. In this study, "accepted" is a label for high-quality documents, and "harmful" and "low quality" are labels for low-quality documents.

The evaluation metrics are accuracy, precision, recall, detection-power, F-score, and ROC-AUC for

[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://huggingface.co/line-corporation/japanese-large-lm-3.6b
[7] https://github.com/HojiChar/HojiChar
[8] https://fasttext.cc/
[9] https://github.com/llm-jp/llm-jp-corpus/tree/main/benchmark

	accepted	harmful	low quality
Number	235	20	245

Table 1: Label distribution of WCQEB.

the binary classification of whether a benchmark document is of high quality (positive example) or low quality (negative example). Detection-power is the percentage of low-quality documents that are correctly classified as low-quality. ROC-AUC is calculated only for classifiers for which prediction probability is available. We use recall and detection-power as our main metrics for method comparison. The method with the highest recall and detection-power is the one that can remove the most low-quality documents without missing the most high-quality documents.

For the implementation of the single / multilayer perception, we use the Python library scikit-learn[10]. To train the single / multilayer perceptron and fastText, we use 5,000 documents each from the BCCWJ (high-quality documents) and the Japanese Web Corpus (low-quality documents) for a total of 10,000 documents. We evaluate the classification results for the benchmark documents. Note that the single-layer perceptron does not provide outputs as probabilities, and thus probability calibration is performed using scikit-learn. To train the language model, we use only sentences ending with a punctuation mark "。" out of the entire BCCWJ corpus. A threshold is set based on the perplexity distribution during inference, and the benchmark documents are classified and evaluated around the threshold. The Transformer-based language model is trained with GPT-NeoX (Andonian et al., 2021) with 19M parameters. The N-gram language model is trained using KenLM (Heafield, 2011), and the 2 to 5-gram language models are compared, with 1-gram as the unit of word segmentation of the morphological analyzer, MeCab[11]. As a preliminary experiment, we also tested the character-based models, but we finally adopted the MeCab segmentation units, which were more accurate.

We also measured the inference speed of each method, quantifying the time required to classify 10,000 documents in Japanese mC4. Measurements were taken three times, and the average was calculated. The experiments were conducted using five cores of an Intel Xeon Gold 6148 Processor. However, pre-processing such as tokenization was not included in the measurement.

[10] https://scikit-learn.org/stable/

[11] https://taku910.github.io/mecab/

4.2 Experimental Results

The evaluation results on WCQEB are shown in Table 2. In the classifier, a document is high quality if its predicted probability exceeds a threshold, while in the language model, it is if its perplexity is below a threshold.

For the classifier-based methods, ROC-AUC exceeds 0.7 for fastText and the multilayer perceptron (MLP). In particular, fastText scores higher than the MLP for recall and detection-power and has better filtering ability. Furthermore, the performance of the 3 and more-gram language models is higher in detection-power than that of fastText by 17.6 points, even though the recall is lower than fastText by only 3 points, which can remove more low-quality documents. The Transformer-based language model has a low detection-power of 0.165 and cannot remove even 20% of low-quality documents. This is lower than the classification performance of all other methods.

Table 2 also shows the classification speed of each method. FastText has the fastest inference speed at 2.41 seconds, followed by perceptron, MLP, and 3-gram language models at approximately 20 seconds. It should be noted that although the Transformer model is relatively slow, taking 29.13 seconds, it can be significantly accelerated through the use of GPUs. With the 3-gram language model, it is calculated to take approximately 5 hours and 30 minutes to process 10 million documents, which can be made even faster by parallelization.

In sum, the filtering method based on the perplexity of the N-gram language model is the best and has high classification ability even with 3-grams. Example benchmark documents and their perplexity of the 3-gram language model are shown in Figure 1.

The document above in Figure 1 contains many alphabets, numbers, symbols, etc., and has a high perplexity of 722,324.09. The document below is out of context but has a low perplexity of 184.16.

We also observe that, in most cases, documents that are seemingly written in fluent and meaningful Japanese have mediocre perplexity, ranging widely between these high and low extremes. This is confirmed by other classification examples, showing that it is difficult to evaluate quality at the context level with the N-gram language model. Therefore,

Method	Details	Acc.	Pre.	Rec.	Det.	F-score	ROC.	threshold p	Speed [s]
Classifier	fastText	0.684	0.615	0.863	0.528	0.718	0.725	0.0005	2.41
	Perceptron	0.634	0.692	0.386	0.850	0.496	0.618	0.5	19.92
	Perceptron (cal)	0.616	0.562	0.794	0.461	0.658	0.693	0.005	20.00
	MLP	0.624	0.565	0.841	0.434	0.676	0.735	0.005	21.35
LM	2-gram LM	0.748	0.707	0.785	0.715	0.744	—	6700	3.18
	3-gram LM	0.764	0.711	0.833	0.704	0.767	—	6700	19.77
	4-gram LM	0.766	0.712	0.837	0.704	0.769	—	6700	29.86
	5-gram LM	0.766	0.712	0.837	0.704	0.769	—	6700	48.77
	Transformer	0.502	0.481	0.888	0.165	0.624	—	60	29.13

Table 2: Evaluation results of each filtering method on WCQEB. "Perceptron (cal)" shows the result of the perceptron after probability calibration.

Figure 1: Perplexity examples of benchmark documents. Both are low-quality Japanese documents with translations by DeepL[12] below.

context-level quality filtering is a future work.

5 Additional Experiments

We conduct additional experiments for detailed performance evaluation through BERT pre-training with a filtered Web corpus and fine-tuning with JGLUE, Japanese General Language Understanding Evaluation. We also analyze how the topics of the Web corpus are affected by the filtering intensity. Furthermore, we show the relevance of URL domain filtering to one based on the n-gram language model.

5.1 Downstream Task Evaluation in JGLUE

Based on the experimental results in Section 4.2, we create a training corpus using the perplexity-based classification method of the 3-gram language model. We create four datasets from a subset of the Japanese mC4 dataset, comprising of approximately 8.5 million documents, by randomly selecting from a subset of 25%, 50%, 75%, and 100% of the entire dataset (8.5 million documents) in order of decreasing value of perplexity. Each dataset consists of 2B tokens, which are tokenized by BertJapaneseTokenizer[13]. The BERT model with 110M parameters is pre-trained on the four datasets, fine-tuned three times for each JGLUE task: MARC-ja, JCoLA, JSTS, JNLI, and JComQA, and the average of the scores is calculated.

Results The evaluation results are shown in Table 3. No model is consistently superior across all JGLUE scores. However, the average score of the lower 25% perplexity filtering (the strongest one), which is the lowest of all, indicates that too much filtering does not improve the performance of the downstream tasks. The models with filtering outperformed those without filtering on some tasks.

5.2 Topic Analysis of a Web Corpus

Topics of each web document vary widely, including news, entertainment, and personal life. There may be biases in the quality of documents depending on the topic. We examine changes in the topic ratio of documents while increasing the filtering strength based on the lower [100, 75, 50, 25]% of perplexity for 100,000 documents in the Japanese mC4 dataset. Using 100,000 unfiltered documents as training data, a topic model is created using LDA (Blei et al., 2003) to calculate the percentage of topics for each document below the perplexity threshold. To make the topic model, the text of the dataset is morphologically analyzed, and only nouns are extracted. From this, numbers, symbols,

[12] https://www.deepl.com/translator

[13] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

Model	MARC-ja/acc	JCoLA/acc	JSTS/pearson	JSTS/spearman	JNLI/acc	JComQA/acc	Average
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

Table 3: JGLUE evaluation results for BERT models with different filtering intensities.

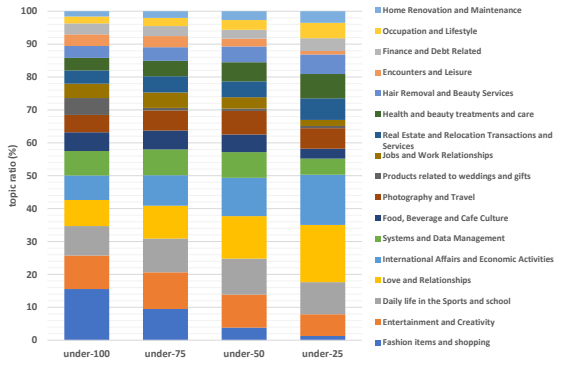

Figure 2: Percentage of topics by filtering strength.

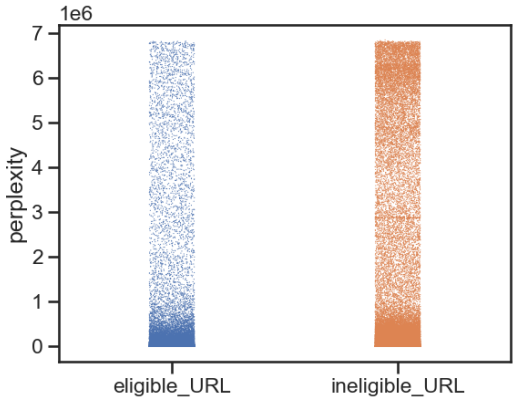

Figure 3: Perplexity distributions for documents with eligible and ineligible URLs.

alphabetical characters, and Japanese stop words[14] are removed, and high-frequency words that occur in more than 30% of the documents are also removed.

Results 17 topics were obtained, and the top 30 most frequently occurring words in each topic were used to name the topics with GPT-3.5[15]. The relationship between the topic proportion and the filtering strength based on the perplexity of the N-gram language model is shown in Figure 2. The document proportion of "fashion items and shopping" has decreased from 15.6% to 1.3% in the filtering process. This topic contains many documents from mail-order sites such as "rakuten.co.jp". Instead of the decrease in the ratio of "Fashion Items and Shopping," the percentage of documents on "International Issues and Economic Activities" and "Love and Relationships" has increased by more than seven percentage points. These mainly include news articles and blog posts. These results indicate a bias in the topics of the Web documents removed by the N-gram language model.

5.3 Comparison between URL Domain and N-gram LM Filters

One of the typical rule-based filtering methods is a URL domain filter, which filters out Web documents with domains other than specific URL domains. LLM-jp's list[16] of eligible URL (top-level) domains consists of ["biz," "cc," "com," "info," "jp," "me," "net," "org," "site," "tokyo," "tv," "work," "xyz"], where URLs that include these top-level domains are considered eligible, while those that do not are ineligible. To compare the URL domain filter with the N-gram language model, we analyze the perplexity distribution of 50,000 Web documents with eligible or ineligible URLs in Japanese mC4.

Results Figure 3 shows the perplexity distributions for documents with eligible and ineligible URLs.

From Figure 3, the perplexity distribution of eligible URLs is concentrated between 0 and 1,000,000. The perplexity distribution of ineligible URLs is focused not only between 0 and 1,000,000 but also between 5,000,000 and 7,000,000. Furthermore, the median perplexity of eligible URLs is 3,299.08, while that of ineligible URLs is 141,572.26. Thus, Web documents with ineligible URLs tend to have a higher perplexity than those with eligible URLs. This result indicates a correlation between the removed documents of the URL domain filter and the N-gram language

[14] http://svn.sourceforge.jp/svnroot/slothlib/CSharp/Version1/SlothLib/NLP/Filter/StopWord/word/Japanese.txt

[15] https://chat.openai.com/

[16] https://github.com/llm-jp/llm-jp-corpus/blob/main/scripts/dict/ja_valid_domains.txt

model. However, the URL domain filter may also remove high-quality documents with low perplexity that have ineligible URLs. In this respect, filtering based on the N-gram language model has an advantage. However, since ineligible URLs account for approximately 7.7 percent of the Japanese mC4 corpus, filtering based on URL domains for the entire corpus has little effect on removing high-quality documents.

6 Conclusion

In this study, we used machine learning-based methods for quality filtering of a Japanese Web corpus and compared their performance on a quality evaluation benchmark. The experimental results showed that the classification method using the perplexity by an N-gram language model had the highest accuracy. However, too much filtering led to performance degradation in downstream tasks. In the future, we plan to evaluate downstream tasks with larger models and consider filtering at more fine-grained units such as paragraphs.

Limitations

This study focused on Japanese web text; however, a future task is to verify whether similar results can be obtained in English to make broader contributions to the field. Furthermore, the approach adopted in this study may introduce biases due to the data used to train classifiers or language models, as illustrated in Section 5.2. Consequently, a more thorough analysis will be required to address these potential biases.

Acknowledgements

We thank Hirokazu Kiyomaru of Kyoto University and LLM-jp for creating and publishing the Web Corpus Quality Evaluation Benchmark. This work was partially supported by JSPS KAKENHI Grant Numbers JP21H04901 and JP24H00727.

References

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. 2021. GPT-NeoX: Large Scale Autoregressive Language Modeling in PyTorch.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*.

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. 2020. The pile: An 800gb dataset of diverse text for language modeling. Abs/2101.00027.

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. 2023. Textbooks are all you need. Abs/2306.11644.

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

Kentaro Kurihara, Daisuke Kawahara, and Tomohide Shibata. 2022. Jglue: Japanese general language understanding evaluation. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2957–2966.

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

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2013. Balanced corpus of contemporary written japanese. *Language Resources and Evaluation*, 48(2):345–371.

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

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

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. 2023. Llama: Open and efficient foundation language models. Abs/2302.13971.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.