Building a Large Japanese Web Corpus for Large Language Models

Naoaki Okazaki[†], **Kakeru Hattori**[†], **Hirai Shota**[†], **Hiroki Iida**[†], **Masanari Ohi**[†],
Kazuki Fujii[†], **Taishi Nakamura**[†], **Mengsay Loem**[†], **Rio Yokota**[‡], **Sakae Mizuki**[†]
[†] Department of Computer Science, School of Computing, Tokyo Institute of Technology
[‡] Global Scientific Information and Computing Center, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550 Japan
{okazaki@c, kakeru.hattori@nlp.c, shota.hirai@nlp.c, hiroki.iida@nlp.c,
masanari.ohi@nlp.c, kazuki.fujii@rio.gsic, taishi.nakamura@rio.gsic,
mengsay.loem@nlp.c, rioyokota@rio.gsic, sakae.mizuki@nlp.c}.titech.ac.jp

Abstract

Open Japanese large language models (LLMs) have been trained on the Japanese portions of corpora such as CC-100, mC4, and OSCAR. However, these corpora were not created for the quality of Japanese texts. This study builds a large Japanese web corpus by extracting and refining text from the Common Crawl archive (21 snapshots of approximately 63.4 billion pages crawled between 2020 and 2023). This corpus consists of approximately 312.1 billion characters (approximately 173 million pages), which is the largest of all available training corpora for Japanese LLMs, surpassing CC-100 (approximately 25.8 billion characters), mC4 (approximately 239.7 billion characters) and OSCAR 23.01 (approximately 74 billion characters). To confirm the quality of the corpus, we performed continual pre-training on Llama 2 7B, 13B, 70B, Mistral 7B v0.1, and Mixtral 8x7B Instruct as base LLMs and gained consistent (6.6–8.1 points) improvements on Japanese benchmark datasets. We also demonstrate that the improvement on Llama 2 13B brought from the presented corpus was the largest among those from other existing corpora.

1 Introduction

ChatGPT, released by OpenAI in late 2022, established a milestone toward achieving general-purpose artificial intelligence. Research topics using large language models (LLMs) attract much attention including chain of thoughts (Wei et al., 2022; Kojima et al., 2022; Wang et al., 2023a; Trivedi et al., 2023), instruction tuning (Wang et al., 2023b;c), evaluation (Suzgun et al., 2023; Zheng et al., 2023), hallucination (Rawte et al., 2023; Zhang et al., 2024), bias (Ladhak et al., 2023; Feng et al., 2023), detecting generated text (Mitchell et al., 2023) watermarking (Kirchenbauer et al., 2023), poisoning (Wan et al., 2023), detecting pre-training data (Shi et al., 2024), unlearning (Yao et al., 2023), and efficient inference (Dettmers et al., 2023).

While there are various motivations, such as raising the level of research and development in natural language processing, elucidating the mechanisms of LLM intelligence, the security risks of relying on a handful of foreign companies, and achieving responsible artificial intelligence, several Japanese companies and universities have actively been developing open LLMs that achieve good performance on Japanese text. However, in many cases, the quality of the training data for building Japanese LLMs was not satisfactory because they were mostly developed overseas as a part of multilingual corpora.

Table 1 lists representative corpora used for training LLMs. Some popular corpora such as Pile-CC (Gao et al., 2020), ROOTS (Laurençon et al., 2023), Dolma (Soldaini et al., 2023), and RedPajama-Data-v2[1] cannot be used for training Japanese LLMs, including no Japanese text.

[1] https://www.together.ai/blog/redpajama-data-v2

<table>
<tr><th>Corpus</th><th>Size (en)</th><th>Size (ja)</th><th>Source</th><th>Text</th><th>Langdet</th></tr>
<tr><td rowspan="3">statmt.org/ngrams
(Buck et al., 2014)</td><td>976 BT</td><td>—</td><td>2012, 2013</td><td>WET?</td><td>cld2</td></tr>
<tr><td colspan="5"><b>Dedup</b>: Remove lines with exact hash values (MurmurHash, 64 bit).</td></tr>
<tr><td colspan="5"><b>Clean</b>: Remove email addresses and HTML fragments; normalize punctuation and capitalization (Moses truecaser).</td></tr>
<tr><td rowspan="3">CC-100 (CCNet)
(Wenzek et al., 2020)</td><td>532 BT</td><td>26 BL</td><td>2018</td><td>WET</td><td>fastText</td></tr>
<tr><td colspan="5"><b>Dedup</b>: Remove paragraphs with exact hash values (SHA-1, 64 bit).</td></tr>
<tr><td colspan="5"><b>Clean</b>: Use documents with high likelihood computed by the language model trained on Wikipedia.</td></tr>
<tr><td rowspan="3">C4
(Raffel et al., 2020)</td><td>156 BT</td><td>—</td><td>April 2019</td><td>WET</td><td>langdetect</td></tr>
<tr><td colspan="5"><b>Dedup</b>: Remove exact matches in three-sentence units.</td></tr>
<tr><td colspan="5"><b>Clean</b>: Keep only lines ending in a punctuation mark; remove documents with one or two sentences; remove lines with 4- words; etc.</td></tr>
<tr><td rowspan="3">mC4
(Xue et al., 2021)</td><td>2,733 BT</td><td>240 BL</td><td>71 months</td><td>WET</td><td>cld3</td></tr>
<tr><td colspan="5"><b>Dedup</b>: Same as C4</td></tr>
<tr><td colspan="5"><b>Clean</b>: Same as C4, but they use a modified rule for keeping text (because the punctuation filter of C4 is specific to English): the text must be at least 200 characters long and contain at least 3 lines.</td></tr>
<tr><td rowspan="3">OSCAR 23.01
(Abadji et al., 2022)</td><td>523 BW</td><td>74 BL</td><td>Nov/Dec 2022</td><td>WET</td><td>fastText</td></tr>
<tr><td colspan="5"><b>Dedup</b>: TLSH</td></tr>
<tr><td colspan="5"><b>Clean</b>: Remove documents with high likelihood computed by the language models trained on harmful content or with URLs registered in Blacklists UT1.</td></tr>
<tr><td rowspan="3">Pile-CC
(Gao et al., 2020)</td><td>2,312 BL</td><td>—</td><td>2013–2020</td><td>jusText</td><td>cld2</td></tr>
<tr><td colspan="5"><b>Dedup</b>: MinHash (Jaccard coefficient threshold is approximately 0.5).</td></tr>
<tr><td colspan="5"><b>Clean</b>: Use documents similar to those in OpenWebText2 (documents with a Reddit score of 3 or higher) based on a fastText classifier.</td></tr>
<tr><td rowspan="3">ROOTS
(Laurençon et al., 2023)</td><td>484 BB</td><td>—</td><td>(539 domains)</td><td>(original)</td><td>fastText</td></tr>
<tr><td colspan="5"><b>Dedup</b>: SimHashLSH (6-gram)</td></tr>
<tr><td colspan="5"><b>Clean</b>: Remove documents with too few words, with a high percentage of repeated characters, with a high percentage of repeated words, with too many emojis, or with too few function words, etc.</td></tr>
<tr><td rowspan="3">RefinedWeb
(Penedo et al., 2023)</td><td>600 BT</td><td>—</td><td>Until Jun 2023</td><td>trafilatura</td><td>fastText</td></tr>
<tr><td colspan="5"><b>Dedup</b>: MinHash (character 5-gram, approximate threshold 0.8) .</td></tr>
<tr><td colspan="5"><b>Clean</b>: Blacklist URL filter by UT1; remove repetitions; per-document filter; per-line modification; remove lines matching more than 50 tokens based on suffix array.</td></tr>
<tr><td rowspan="3">Dolma
(Soldaini et al., 2023)</td><td>2,415 BT</td><td>—</td><td>May 2020 to Jun 2023</td><td>WET</td><td>fastText</td></tr>
<tr><td colspan="5"><b>Dedup</b>: Remove documents with identical URLs; remove identical paragraphs within documents.</td></tr>
<tr><td colspan="5"><b>Clean</b>: Paragraph filtering of MassiveWeb (Rae et al., 2021); punctuation rules of C4; etc.</td></tr>
<tr><td rowspan="2">ClueWeb 22
(Overwijk et al., 2022)
(Non-commercial use)</td><td>*16.7 TT</td><td>3,301 BT</td><td>Search engine</td><td>BlingFire</td><td>(original)</td></tr>
<tr><td colspan="5"><b>Dedup</b> and <b>Clean</b>: Documents were sampled from a commercial search engine based on the distribution of page visits.</td></tr>
<tr><td rowspan="3">RedPajama-Data-v2</td><td>20.5 TT</td><td>—</td><td>All (84 dumps)</td><td>WET</td><td>fastText</td></tr>
<tr><td colspan="5"><b>Dedup</b>: Same as CCNet</td></tr>
<tr><td colspan="5"><b>Clean</b>: Text-quality estimation rules used by C4, RefinedWeb, etc.</td></tr>
<tr><td rowspan="2">This study (raw)</td><td>3,684 BL</td><td>3,684 BL</td><td>2020-40 to 2023-23</td><td>trafilatura</td><td>(original)</td></tr>
<tr><td colspan="5"><b>Dedup</b> and <b>Clean</b>: None.</td></tr>
<tr><td rowspan="3">This study (clean)</td><td>312 BL</td><td>312 BL</td><td>2020-40 to 2023-23</td><td>trafilatura</td><td>(original)</td></tr>
<tr><td colspan="5"><b>Dedup</b>: MinHash (character 5-gram, Jaccard coefficient threshold is approximately 0.9).</td></tr>
<tr><td colspan="5"><b>Clean</b>: Repetition detection of RefinedWeb and quality filter specially designed for Japanese text.</td></tr>
</table>

Table 1: Representative corpora that can be used to pre-train LLMs. BL, BW, BT, and TT stand for billion letters, billion words, billion tokens, and trillion tokens, respectively. Extracted from the corresponding papers, numbers in English size are not directly comparable because of the difference in units. The number of ClueWeb22 may include text in other languages. Numbers in Japanese size are in Unicode characters and are directly comparable. **Text**, **Langdet**, **Dedup**, and **Clean** explain methods for text extraction, language detection, deduplication, and cleaning. WET in text extraction indicates that the corpus uses the text extraction results distributed by Common Crawl in WET format.

Therefore, CC-100 (Wenzek et al., 2020), mC4 (Xue et al., 2021), and OSCAR 23.01 (Abadji et al., 2022) are candidates for training Japanese LLMs, including a certain amount of Japanese text and released with permissive licenses. However, the quality of Japanese text in these corpora is unsatisfactory because they contain noise in the HTML-to-text conversion (Common Crawl WET format) and because they have incorporated no special efforts to improve text quality for Japanese. Although we recognize the importance of efforts in building multilingual corpora, it is difficult to build a useful and reliable Japanese corpus without the knowledge of Japanese language. Concurrently to our work, Enomoto et al. (2024) explored filtering methods for Japanese web corpora, but this study did not demonstrate the effectiveness of the filtering methods on generative LLMs.

Therefore, this paper explores a method to construct a large-scale, high-quality Japanese web corpus that can be used for training Japanese LLMs. The presented method includes lightweight language detection that boosts the processing speed of text extraction in the target language. This strategy is applicable not only to Japanese but also to other languages with less text than English. In addition, we specially design a filtering method to find good-quality Japanese text. The corpus developed in this study is made from 21 snapshots of Common Crawl (from CC-MAIN-2020-40 to CC-MAIN-2023-23), and the size of the corpus after cleaning is 173,350,375 pages and 312,093,428,689 characters. To confirm the quality of the corpus, we perform continual pre-training[2] on Llama 2 7B, 13B, 70B (Touvron et al., 2023), Mistral 7B v0.1 (Jiang et al., 2023), and Mixtral 8x7B Instruct (Jiang et al., 2024) as base LLMs. Experimental results demonstrate that continual pre-training consistently improves the base model's performance by 6.6–8.1 points on Japanese benchmark datasets. We also demonstrate that the improvement on Llama 2 13B brought from the presented corpus was the largest among those from other existing corpora. The models trained on the presented corpus are available on the web site[3] and Hugging Face[4]. The implementations of this study are available on GitHub[5][6].

2 Related work

A number of large corpora were built from Common Crawl[7] archives. Common Crawl is a non-profit organization that crawls websites and provides their archives. The crawled data was initially distributed in ARC format[8], but since the summer of 2013, it has been stored in Web ARChive (WARC)[9] and Web Text (WET) formats. According to Statistics of Common Crawl Monthly Archives[10], the total amount of accessible archives (data crawled since 2013 till 2023) is 251,325,355,174 pages. Compact Language Detector 2 (`cld2`)[11] estimated that about 5 percent of these web pages are written in Japanese.

Table 1 summarizes the corpora derived from Common Crawl (except for ClueWeb22, where documents are sampled from a commercial search engine). All permissive corpora that contain Japanese text rely on the data in WET format (HTML pages were converted to text by Common Crawl). The advantage of processing data in WET format is the reduction of processing time and data transfer size. Wenzek et al. (2020) proposed a pipeline to extract multilingual text from Common Crawl to train a multilingual BERT model (XLM-

[2] Continual pre-training on a base model starts with the weights (parameters) of the base model and performs another round of pre-training on new data (the Japanese corpus in this study). This is also called as *continued pretraining*.

[3] `https://swallow-llm.github.io/`

[4] `https://huggingface.co/tokyotech-llm`

[5] The entire pipeline except for deduplication: `https://github.com/swallow-llm/swallow-corpus`

[6] Deduplication: `https://github.com/swallow-llm/doubri`

[7] `https://commoncrawl.org/`

[8] `https://archive.org/web/researcher/ArcFileFormat.php`

[9] `https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1/`

[10] `https://commoncrawl.github.io/cc-crawl-statistics/`

[11] `https://github.com/CLD2Owners/cld2`

R) (Conneau et al., 2020), and released its implementation (CCNet)[12] and data (CC-100)[13]. Xue et al. (2021) constructed mC4, a multilingual extension of C4 (Raffel et al., 2020) to train mT5, a multilingual variant of T5. Abadji et al. (2022) released OSCAR 23.01 with adult content filtering (based on an *n*-gram language model) and duplicate removal (based on locality sensitive hashing) added to the previous release.

However, the corpus construction procedure often introduces aggressive filtering rules because the WET data usually includes irrelevant text, e.g., noises in HTML-to-text conversion, JavaScript codes, navigation menus, and footers. For example, C4 (Raffel et al., 2020) had to introduce filtering rules such as "remove lines with the word *JavaScript*" (to remove JavaScript code) and "remove pages with curly brackets `{}`" (because curly brackets are rarely used in a natural language but often used in a programming language). Therefore, we use WARC instead of WAT as the source format of Common Crawl archives.

We follow the design principle of RefinedWeb (Penedo et al., 2023): *scale first* (avoid labour intensive human curation process), *strict deduplication* (respect the value of deduplication for large language models), and *neutral filtering* (avoid undesirable biases introduced by machine-learning-based filtering). The construction procedure of RefinedWeb has much in common with MassiveWeb (Rae et al., 2021). Collecting web pages with their own web crawler, Rae et al. (2021) used Google's SafeSearch filter to remove harmful content, and extracted text based on the HTML DOM structure. They also removed documents that contained a lot of repetitions, low-quality text, or duplicated information. In this study, we extend their pipelines to extract high-quality Japanese documents efficiently.

We also considered using ClueWeb22 (Overwijk et al., 2022) as a source for building a large Japanese corpus. Although ClueWeb22 has nice and unique properties such as real distribution (web pages are extracted by the crawler of a commercial search engine and follow the distribution of web search activities), large-scale quality content (content extraction pipeline based on the search engine's production-quality content understanding system), and rich information (annotations of content structure, in-links/out-links of web pages, visual features of contents, etc), we could not use ClueWeb22 in this study because an LLM trained on ClueWeb22 cannot be released under an open license.

3 Method

Figure 1 shows the corpus building pipeline. This pipeline is roughly divided into the following three stages: (1) Extracting Japanese text from WARC files in Common Crawl (Section 3.1); (2) Selecting Japanese text carefully with quality filtering (Section 3.2), deduplication (Section 3.3), and host filtering (Section 3.4); (3) Cleaning extracted text (Section 3.5).

3.1 Extracting Japanese text

Common Crawl snapshots are stored as buckets in Amazon S3 and can be accessed via S3 or the web server[14]. We extract HTML content from WARC data using the `WARCIO` library[15] (Step 1, Figure 1). We will defer to the explanation of Step 2 until later. We then apply `Trafilatura`[16] (Barbaresi, 2021) for extracting text from HTML markups (Step 3). Step 4 detects the language of a text based on a linear binary classifier (see Appendix A.1 for the detail) and extracts Japanese text only.

Because about 5% of the entire Common Crawl is written in Japanese, it is possible to reduce the processing time of Steps 3 and 4 by 95% if we can target Japanese web pages only. However, in order to improve the accuracy of language detection, it is desirable to remove HTML markup before applying the language detection. Therefore, we should apply text

[12] `https://github.com/facebookresearch/cc_net`

[13] `https://huggingface.co/datasets/cc100`

[14] `https://data.commoncrawl.org/`

[15] `https://github.com/webrecorder/warcio`

[16] `https://trafilatura.readthedocs.io/`

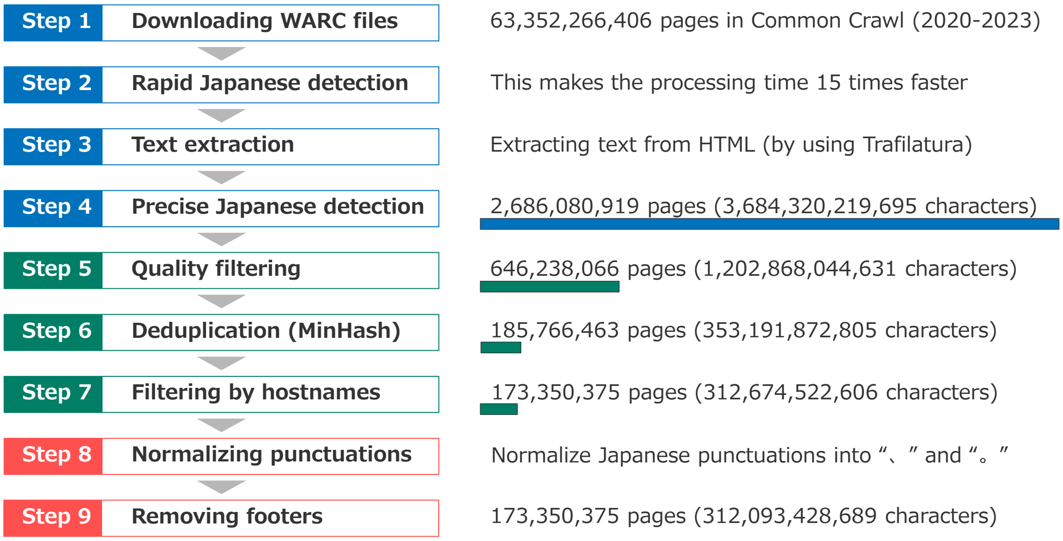
Step 1
Downloading WARC files
63,352,266,406 pages in Common Crawl (2020-2023)
Step 2
Rapid Japanese detection
This makes the processing time 15 times faster
Step 3
Text extraction
Extracting text from HTML (by using Trafilatura)
Step 4
Precise Japanese detection
2,686,080,919 pages (3,684,320,219,695 characters)
Step 5
Quality filtering
646,238,066 pages (1,202,868,044,631 characters)
Step 6
Deduplication (MinHash)
185,766,463 pages (353,191,872,805 characters)
Step 7
Filtering by hostnames
173,350,375 pages (312,674,522,606 characters)
Step 8
Normalizing punctuations
Normalize Japanese punctuations into “、” and “。”
Step 9
Removing footers
173,350,375 pages (312,093,428,689 characters)

Figure 1: The pipeline for building a large Japanese web corpus.

extraction first (Step 3), followed by language detection (Step 4). To make matters worse, text extraction takes more processing time than language detection; we cannot reduce the processing time in this processing order.

This is the motivation behind Step 2: we select web pages that are highly likely to be in Japanese in a rapid manner without text extraction. A web page proceeds to Steps 3 and 4 only when one of the following two conditions is met: (1) the HTML declares the language as Japanese, for example, `<html lang="ja">`; (2) the precise Japanese detection (Section A.1) recognizes the content within `<title>` tag as Japanese.

To evaluate the validity of this rule, we compared the results of rapid and precise language detection. More specifically, assuming that the result of the precise language detection is correct, we measured the accuracy of rapid language detection on 10 WARC files[17] (394,192 pages, 12,052,992,707 bytes in total with gzip compression). The precision, recall, and F1 score were 0.888, 0.967, and 0.926, respectively. This result indicates that although rapid Japanese language detection may discard about 3.3% of Japanese web pages, the relatively high precision reduces the processing time for Steps 3 and 4 for non-Japanese web pages. In a benchmark experiment using 1 CPU of Intel Xeon Gold 6130 processor (2.1 GHz), the processing time for Steps 1–4 was about 15 times faster than Steps 1, 3, and 4 (without rapid language detection) because Step 2 removes a number of web pages from the target.

This stage obtained 2,686,080,919 pages and 3,684,320,219,695 Japanese characters from 21 snapshots of Common Crawl (from CC-MAIN-2020-40 to CC-MAIN-2023-23).

3.2 Quality filtering

This process (1) removes web pages with many repetitions, (2) selects web pages that contain good-quality Japanese text, and (3) removes web pages that may contain harmful expressions. For processing (1), we adopted the rules of Rae et al. (2021) to remove documents with many duplicated contents (see Appendix A.2).

As for processing (2), no standard has been established for assessing the quality of Japanese text. Therefore, we designed several rules for (2) selecting Japanese text of good quality. This study removes a text that satisfies either of the following conditions as irrelevant for LLMs (thresholds in parentheses).

[17] CC-MAIN-20230527223515-20230528013515-0000?.warc.gz from 0 to 9 in CC-MAIN-2023-23.

1. Number of characters (less than 400 characters)
2. Percentage of hiragana characters (less than 0.2)
3. Percentage of katakana characters (greater than 0.5)
4. Percentage of Japanese characters (hiragana, katakana, kanji, punctuations) (less than 0.5)
5. Average number of characters in a sentence (less than 20 or more than 90)
6. Number of characters in the longest sentence (more than 200)
7. Percentage of sentences ending with an ellipsis symbol (greater than 0.2)

For example, Rule 2 ensures that a text includes a certain amount of function words such as *ga* (subject case marker), *wo* (object case marker) *ni* (position case maker roughly corresponding to *to* in English). Rule 3 rejects a web page containing a lot of product or service names (e.g., found in SEO sites), which are often expressed in katakana characters. Rule 7 removes a web page that looks like an RSS feed (a collection of snippets of other web pages). These rules were adjusted manually as the authors examined the text to be deleted and extracted.

Although RefinedWeb used UT1 blocklist[18] for removing harmful content, its coverage for Japanese pages may be insufficient. In order to remove web pages that may contain harmful expressions (3), we manually created a list of "NG expressions," which may indicate harmful (e.g., adult, violent, aggressive) contents. If the percentage of characters that match NG expressions is greater than 0.05, we exclude the text from the corpus.

This quality filtering reduced the size of the corpus to 646,238,066 pages (1,202,868,044,631 characters). Examining the text before and after the quality filtering, we observe that web pages that may be unuseful for training LLMs (e.g., e-commerce sites) have disappeared.

3.3 Deduplication

Because Common Crawl visits and crawls the same website multiple times, the archive includes web pages that are identical or similar because of minor modifications or reproductions. Lee et al. (2022) reported that deduplication, removing duplicated text from the corpus, not only reduces memorization of LLMs but also improves pre-training efficiency.

Therefore, we performed deduplication using MinHash (Broder, 1997) in a similar manner to Lee et al. (2022). When using MinHash for deduplication, it is common to create r buckets, where each bucket is a concatenation of b MinHash values, compare r pairs of buckets of two documents, recognize the two documents as a duplicate if any of bucket pairs are identical. In this study, we adopted a setting where $b = 20, r = 40$ so that a pair of documents with a Jackard coefficient[19] of 0.9 can be approximately detected as a duplicate with a probability of 92.5%. When a pair of documents was recognized as a duplicate, we kept the one crawled recently and removed the older one. In the snapshot used in this experiment, the non-duplicate rate of web pages collected between March and June 2023 ranged from 77.8 to 87.9 percent, while the non-duplicate rate of web pages collected before February 2023 dropped to less than 40 percent, and the rate of web pages collected around 2020 was less than 20 percent. This deduplication process reduced the corpus size to 185,766,463 pages (353,191,872,805 characters).

3.4 Filtering by hostnames

Even after the quality filtering in Section 3.2, we found some irrelevant content in the corpus. Therefore, we created a block list[20] consisting of hostnames that met the following criteria.

[18] `https://dsi.ut-capitole.fr/blacklists/`

[19] In this study, features were constructed with 5-grams of characters.

[20] We used the word blocklist as "a list for blocking," following the name "UT1 blocklist." We are reluctant to use the word "black list" because some hosts in this list may not be "black." For example, a hospital website explaining a woman's maternity is prone to be included in the list.

1. Included in the UT1 blocklist.
2. Percentage of pages containing the name of a dating site (greater than 0.001).
3. Percentage of pages containing NG expressions (greater than 0.005).
4. `*wikipedia.org`
5. `*.5ch.net`

Although the UT1 blocklist includes some Japanese websites, we introduced Criteria 2 and 3 to improve the coverage. The authors manually adjusted the thresholds for the percentage[21]. We applied Criterion 4 because we planned to use Wikipedia text extracted from the dump. This filtering process resulted in a corpus of 173,350,375 pages (312,674,522,606 characters).

3.5 Cleaning extracted text

Sections 3.1 to 3.4 process text at the document level, i.e., filter out irrelevant documents without altering the text within a document. In this study, to avoid an unexpected side effect, we minimally edit the text: punctuation normalization (replacing Western-style punctuations with Japanese-style ones) and footer trimming (removing footers that were left by `Trafilatura`). Refer to Appendix A.3 for the details. The process does not delete web pages, but the number of characters decreased from 312,674,522,606 to 312,093,428,689.

4 Experiments

4.1 Models and training data

In order to evaluate the usefulness of the presented corpus, we conducted continual pre-training of popular LLMs that excel in English[22]. We used Llama 2 7B, 13B, 70B, Mistral 7B v0.1, and Mixtral 8x7B Instruct v0.1 as base models and performed continual pre-training of these base models on the presented corpus to examine whether the corpus improves their Japanese knowledge and skill. The training data for continual pre-training was a mixture of the presented corpus, Japanese Wikipedia, RefinedWeb, and arXiv component in The Pile[23] (Gao et al., 2020). Specifically, we prepared approximately 104.9 billion tokens of training data, assuming a sequence length of 4,096 tokens and a batch size of 1,024 with 25,000 steps for continuous pre-training. The ratio of Japanese to English tokens was set to 9:1, with 5% of the training data being the English text from RefinedWeb, 5% being The Pile's arXiv paper text (English), and the remaining 90% being Japanese text. The breakdown of the Japanese text was about 1.6 billion tokens of Japanese Wikipedia, and the presented Japanese corpus occupied the rest. We also used AlgebraicStack (Azerbayev et al., 2024) for Mistral's continual pre-training. For Mixtral, we used both AlgebraicStack and The Vault (Manh et al., 2023)[24]. We used a different ratio (72:28) of Japanese to English tokens for continual pre-training on Mixtral 8x7B Instruct[25].

We did not change the architecture of base LLMs; the embedding sizes of tokens, hidden layers, feed-forward layers, the number of attention heads, and the number of layers were

[21] We placed emphasis on recall at the expense of precision to reduce harmful behaviors of trained models. This design results in including a lot of non-harmful websites in the block list.

[22] It may be straightforward to evaluate the corpus by training LLMs from scratch, but we chose continual pre-training because the broader goal of this effort was to build high-performance LLMs that excel in Japanese under the limited amount of computing budget. We can assess the quality of the corpus by checking performance gains from the base models and also comparing the performance with other models.

[23] We expected that the RefinedWeb and arXiv would help maintain the base LLMs' English knowledge and skill during continual pre-training.

[24] We completed the continual pre-training experiments on Llama 2 well before we began with Mistral 7B. By then, our objective had expanded to enhance logical inference capabilities in Mistral and Mixtral, leading us to incorporate AlgebraicStack and The Vault.

[25] We changed this ratio as an attempt to improve the performance in English.

unchanged from the base models in continual pre-training. Before continual pre-training, we added Japanese vocabulary to Llama 2 and Mistral 7B tokenizers. The total number of vocabulary was 43,176 for LLMs based on Llama 2 and 42,800 for those based on Mistral 7B. We employed AdamW (Loshchilov & Hutter, 2019), with hyperparameters set to $\beta_1 = 0.9$, $\beta_2 = 0.95$, $\epsilon = 1.0 \times 10^{-8}$. We used re-warming and re-decaying the learning rate (Ibrahim et al., 2024) with 1,000 warm-up steps. For Llama 2, the maximum learning rate was set at 1.0×10^{-4} with a decay rate of $1/30$, while for Mistral and Mixtral, it was 2.0×10^{-5} with a decay rate of $1/10$. The batch size was 1,024. We used 0.1 for weight decay and 1.0 for gradient clipping. In addition, we used Flash Attention (Dao et al., 2022) to improve computational and memory efficiency.

In order to examine the impact of a corpus on the performance of LLMs, we performed continual pre-training on ClueWeb22 and llm-jp-corpus v1.0.1[26] as well as the presented corpus. We also compared our LLMs to others with similar numbers of parameters: CyberAgentLM2 7B (CALM2 7B, a Japanese LLM trained from scratch); Japanese-StableLM-Base-Beta-7B (JSLMB 7B, a continual pre-training LLM on Llama 2 7B); Youri 7B (a continual pre-training LLM on Llama 2 7B); Qwen 7B; Nekomata 7B (a continual pre-training LLM on Qwen 7B); Japanese Stable LM Base Gamma 7B (JSLMG 7B, a continual pre-training LLM on Mistral 7B); Qwen 14B; KARAKURI LM 70B (Karakuri 70B, a continual pre-training LLM on Llama 2 70B); Japanese-StableLM-Base-Beta-70B (JSLMB 70B, a continual pre-training LLM on LLama 2 70B); and Qwen 72B. Refer to Appendix A.4 for the complete list of URLs of the LLMs used in the experiments.

4.2 Evaluation Dataset

We used llm-jp-eval[27] and lm-evaluation-harness[28] as Japanese evaluation benchmarks. llm-jp-eval is a benchmark consisting of tasks for multiple choice (MC) question answering, open question answering (QA), reading comprehension (RC), and natural language inference (NLI) tasks. JCommonsenseQA (Kurihara et al., 2022) is employed for MC, JEMHopQA (Ishii et al., 2023) and NIILC (Sekine, 2003) for QA, JSQuAD (Kurihara et al., 2022) for RC. We decided to exclude the NLI dataset from the benchmark because the distribution of ground-truth NLI labels is highly imbalanced, which made the evaluation unstable[29]. For lm-evaluation-harness, we used Japanese subsets of XL-Sum (Hasan et al., 2021) for the automatic summarization task and MGSM (Shi et al., 2023) for the arithmetic reasoning task, respectively. In addition, WMT 2020 (Barrault et al., 2020) was used for the Japanese-English and English-Japanese machine translation.

4.3 Results

Table 2 reports the performance of LLMs on the Japanese benchmark datasets. The row "+ *this study*" shows the performance of the LLM after we applied continual pre-training to the base model on the presented corpus. The rows "+ *this study*" demonstrate that the presented corpus consistently improves the performance of the base models via continual pre-training by 6.6–8.1 points on average. The models built by this study outperform other models built from scratch or continual pre-training, establishing the state-of-the-art performance in each model size (7B, 13B, 8x7B, and 70B). This fact shows the usefulness of the presented corpus in building LLMs that excel in Japanese.

We can compare different Japanese corpora in continual pre-training of Llama 2 13B. All corpora (ClueWeb22, llm-jp-corpus v1.0.1, and presented corpus) improved the base model's performance by 5.0–7.0 points, which verifies the effectiveness of the continual pre-training. The presented corpus yielded the best improvement (7.0 points) of all corpora; in particular, it drastically enhances the performance of question answering (JCom, JEMHop, NIILC),

[26] `https://github.com/llm-jp/llm-jp-corpus`

[27] `https://github.com/llm-jp/llm-jp-eval` v1.0.0

[28] `https://github.com/Stability-AI/lm-evaluation-harness/tree/jp-stable` commit #9b42d41

[29] An LLM may obtain a high score only if the model happens to predict the majority label without understanding and solving the NLI task.

Corpus	QA	QA	QA	RC	Sum	Ja-En	En-Ja	Math	Avg
	JCom	JEMHop	NIILC	JSQuAD	XL-Sum	WMT20	WMT20	MGSM	
CALM2 7B	21.98	50.47	50.66	77.99	2.33	14.99	23.45	6.00	30.98
JSLMB 7B	36.10	44.78	44.32	83.18	21.95	12.26	19.46	7.20	33.66
Youri 7B	46.20	47.76	49.99	85.06	19.57	**19.71**	26.71	6.40	37.67
Llama 2 7B	38.52	42.40	34.10	79.17	19.05	17.37	17.83	7.60	32.01
+ *this study*	48.08	**50.78**	**59.68**	85.73	18.30	15.11	25.10	12.40	39.40
Qwen 7B	77.12	42.34	23.76	85.94	13.71	18.01	16.89	21.60	37.42
Nekomata 7B	74.17	49.28	50.22	87.07	16.76	18.15	**26.73**	12.40	41.85
JSLMG 7B	73.64	46.43	55.68	**89.10**	**22.93**	15.61	23.90	16.80	43.01
Mistral 7B	73.01	42.45	27.22	85.63	20.06	17.33	14.05	17.60	37.17
+ *this study*	**85.70**	49.15	55.19	88.02	19.88	16.67	24.94	**22.40**	**45.24**
Llama 2 13B	68.19	44.55	41.74	85.51	21.33	19.81	21.36	13.20	39.46
+ ClueWeb22	76.76	49.29	56.02	89.55	20.15	**23.32**	**29.08**	11.60	44.47
+ llm-jp	74.71	**50.85**	60.34	**90.28**	**21.91**	17.89	25.89	18.00	44.98
+ *this study*	**78.37**	50.63	**64.06**	90.07	21.68	17.71	27.37	**21.60**	**46.44**
Qwen 14B	88.29	42.43	32.20	89.80	18.51	22.24	22.23	38.80	44.31
Mixtral 8x7B	84.00	50.33	31.07	88.08	20.02	20.63	19.56	**45.20**	44.86
+ *this study*	**92.58**	**58.43**	**56.87**	**91.48**	**25.89**	**20.74**	**27.05**	43.60	**52.08**
Karakuri 70B	85.79	51.25	57.13	91.00	14.64	21.13	25.40	27.20	46.69
JSLMB 70B	91.15	49.25	60.42	**91.92**	**25.73**	23.35	27.65	41.60	51.38
Llama 2 70B	86.86	46.56	52.56	90.80	23.61	**23.98**	26.43	35.60	48.30
+ *this study*	**93.48**	**62.90**	**69.60**	91.76	22.66	22.98	**30.43**	**48.40**	**55.28**
Qwen 72B	92.94	55.66	45.18	91.59	21.79	23.56	25.61	63.20	52.44

Table 2: Benchmark evaluation on Japanese tasks. A horizontal line groups LLMs with the same number of parameters. A horizontal dash line groups LLMs from the same base model. A bold number indicates the maximum value in the LLMs with the same number of parameters.

reading comprehension (JSQuAD), and arithmetic reasoning (MGSM). Although we did not incorporate a special effort to improve these tasks, the model acquired knowledge about Japan and the Japanese language from the presented corpus. In contrast, the presented corpus did not improve the summarization (XL-Sum) task. This trend is observed across other base models, except for Mixtral 8x7B Instruct, potentially indicating that adding Japanese vocabulary may have a detrimental effect on this task. In addition, we found that the continual pre-training on the presented corpus improved the performance in English-Japanese translation but degraded that in the reversed direction (Japanese-English translation). We will explore mitigation of this phenomenon in the future, e.g., by changing the ratio of Japanese to English tokens in the training data and promoting language transfer using an English-Japanese parallel corpus during continual pre-training.

5 Conclusion

In this paper, we built a large Japanese web corpus by extracting and refining text from the Common Crawl archive (21 snapshots of approximately 63.4 billion pages crawled between 2020 and 2023). This corpus consists of approximately 312.1 billion characters (approximately 173 million pages), which is the largest of all available training corpora for Japanese LLMs. We confirmed the usefulness of the corpus by performing continual pre-training on Llama 2 7B, 13B, 70B, Mistral 7B v0.1, and Mixtral 8x7B Instruct as base LLMs. The experiments demonstrated consistent (6.6–8.1 points) improvements in Japanese benchmark datasets, and established the state-of-the-art performance in each model size (7B, 13B, 8x7B, and 70B). We also observed that the improvement on Llama 2 13B brought from the presented corpus was the largest among those from other existing corpora.

Future directions include efforts towards the safety of LLMs, such as reducing harmful generations (e.g., discrimination, exclusion, toxicity, hallucination). Currently, we only use the lists of NG expressions and hostnames, but it is desirable to establish more robust filtering methods to remove harmful text for pre-training Japanese LLMs. In addition, although our study focused on the continual pre-training setting, we want to evaluate the presented corpus by training Japanese LLMs from scratch. Although we evaluated the LLMs in downstream tasks such as question-answering and summarization, it is questionable whether this can measure the "general intelligence" of an LLM. At the same time, training an LLM on a pre-training corpus requires huge computations. Therefore, we want to explore a lightweight method for assessing the effectiveness of pre-training corpora without building LLMs.

6 Limitations

We could not present an ablation study of each step in Figure 1 because of a large amount of computational resources required to train LLMs. Instead, we reported the performance of each step of the corpus construction procedure in Section 3.

This study focuses on Japanese, and the construction of corpora for other languages is outside the scope. For English, it may be a good strategy to make use of good quality corpora that have already been developed. Although some ideas in this paper could be helpful in building corpora for other languages, for example, reducing processing time with rapid language detection, we believe that it would be better for researchers from different countries to share the task of building corpora for their own, as assessing text quality requires the knowledge of the language.

7 Ethics statement

Article 30-4 of the Copyright Act in Japan permits to use of a copyrighted work without the permission of the copyright holder as long as the use is not a person's purpose to personally *enjoy* or cause another person to *enjoy* the work. This is a quote from the Japanese translation of Article 30-4 of Copyright Act[30]:

> It is permissible to exploit a work, in any way and to the extent considered necessary, in any of the following cases, or in any other case in which it is not a person's purpose to personally enjoy or cause another person to enjoy the thoughts or sentiments expressed in that work; provided, however, that this does not apply if the action would unreasonably prejudice the interests of the copyright owner in light of the nature or purpose of the work or the circumstances of its exploitation: *(snip)*

The nuance of *enjoy* probably needs clarification. *Enjoying* a work roughly corresponds to consuming the idea of a work (for any purpose, including entertainment, learning, and communication, regardless of the monetary gains from the work). The intention of Article 30-4 is to define an exceptional use case where a computer program consumes works directly for machine learning and information analysis, whereas a human cannot do that.

This provides the justification for building the presented corpus and LLMs. To the best of our knowledge, we have taken every measure to keep the corpus as non-toxic and unbiased as possible, but we are unaware of any direct ethical consequences caused by the LLMs trained on the presented corpus.

8 Reproducibility statement

All models developed in this study (continual pre-training on Llama 2 7B, 13B, 70B, Mistral 7B v0.1, and Mixtral 8x7B Instruct) have already been released on Hugging Face. The bench-

[30] https://www.japaneselawtranslation.go.jp/ja/laws/view/4207#je_ch2sc3sb5at4

mark datasets used in this study are also publicly available. Therefore, it is straightforward to reproduce our experimental results reported in Table 2.

Acknowledgements

This paper is based on results obtained from a project, JPNP18002, commissioned by the New Energy and Industrial Technology Development Organization (NEDO). In addition, the experiments of continual pre-training of LLMs was supported by the "Support Program for Building Large Language Models" of the AI Bridging Cloud Infrastructure (ABCI) developed and operated by the National Institute of Advanced Industrial Science and Technology (AIST). We used the datasets and findings released by the Japanese LLM Study Group (LLM-jp) in the evaluation experiments. We also received suggestions on tokenization from Tatsuya Hiraoka of MBZUAI.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 4344–4355, June 2022. URL https://aclanthology.org/2022.lrec-1.463.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In *the Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4WnqRR915j.

Adrien Barbaresi. Trafilatura: A web scraping library and command-line tool for text discovery and extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pp. 122–131, August 2021. doi: 10.18653/v1/2021.acl-demo.15. URL https://aclanthology.org/2021.acl-demo.15.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pp. 1–55, November 2020. URL https://aclanthology.org/2020.wmt-1.1.

Andrei Z Broder. On the resemblance and containment of documents. In *Proceedings of the Compression and Complexity of Sequences 1997*, pp. 21, 1997. ISBN 0818681322. doi: 10.1109/SEQUEN.1997.666900.

Christian Buck, Kenneth Heafield, and Bas van Ooyen. N-gram counts and language models from the Common Crawl. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, pp. 3579–3584, May 2014. URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/1097_Paper.pdf.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, July 2020. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, volume 35, pp. 16344–16359, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/67d57c32e20fd0a7a302cb81d36e40d5-Paper-Conference.pdf.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

Rintaro Enomoto, Arseny Tolmachev, Takuro Niitsuma, Shuhei Kurita, and Daisuke Kawahara. Investigating web corpus filtering methods for language model development in Japanese. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pp. 154–160, June 2024. doi: 10.18653/v1/2024.naacl-srw.18. URL https://aclanthology.org/2024.naacl-srw.18.

Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair NLP models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11737–11762, July 2023. doi: 10.18653/v1/2023.acl-long.656. URL https://aclanthology.org/2023.acl-long.656.

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. arXiv:2101.00027, 2020.

Tahmid Hasan, Abhik Bhattacharjee, Md. Saiful Islam, Kazi Mubasshir, Yuan-Fang Li, Yong-Bin Kang, M. Sohel Rahman, and Rifat Shahriyar. XL-sum: Large-scale multilingual abstractive summarization for 44 languages. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4693–4703, August 2021. doi: 10.18653/v1/2021.findings-acl.413. URL https://aclanthology.org/2021.findings-acl.413.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=DimPeeCxKO.

Ai Ishii, Naoya Inoue, and Satoshi Sekine. Construction of a Japanese multi-hop QA dataset for QA systems capable of explaining the rationale. In *the 29th Annual Meeting of Japanese Association for Natural Language Processing (NLP2023)*, pp. 2088–2093, March 2023. URL https://www.anlp.jp/proceedings/annual_meeting/2023/pdf_dir/Q8-14.pdf.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. arXiv:2310.06825, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. arXiv:2401.04088, 2024.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 17061–17084, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

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, June 2022. URL `https://aclanthology.org/2022.lrec-1.317`.

Faisal Ladhak, Esin Durmus, Mirac Suzgun, Tianyi Zhang, Dan Jurafsky, Kathleen McKeown, and Tatsunori Hashimoto. When do pre-training biases propagate to downstream tasks? a case study in text summarization. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 3206–3219, May 2023. doi: 10.18653/v1/2023.eacl-main.234. URL `https://aclanthology.org/2023.eacl-main.234`.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, et al. The BigScience ROOTS corpus: A 1.6TB composite multilingual dataset. arXiv:2303.03915, 2023.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, May 2022. doi: 10.18653/v1/2022.acl-long.577. URL `https://aclanthology.org/2022.acl-long.577`.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

Dung Nguyen Manh, Nam Le Hai, Anh T. V. Dau, Anh Minh Nguyen, Khanh Nghiem, Jin Guo, and Nghi D. Q. Bui. The vault: A comprehensive multilingual dataset for advancing code understanding and generation. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pp. 219–244, December 2023. doi: 10.18653/v1/2023.nlposs-1.25. URL `https://aclanthology.org/2023.nlposs-1.25`.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24950–24962, 23–29 Jul 2023. URL `https://proceedings.mlr.press/v202/mitchell23a.html`.

Arnold Overwijk, Chenyan Xiong, Xiao Liu, Cameron VandenBerg, and Jamie Callan. ClueWeb22: 10 billion web documents with visual and semantic information. arXiv:2211.15848, 2022.

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. arXiv:2306.01116, 2023.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray

Kavukcuoglu, and Geoffrey Irving. Scaling language models: Methods, analysis & insights from training Gopher. arXiv:2112.11446, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Vipula Rawte, Swagata Chakraborty, Agnibh Pathak, Anubhav Sarkar, S.M Towhidul Islam Tonmoy, Aman Chadha, Amit Sheth, and Amitava Das. The troubling emergence of hallucination in large language models - an extensive definition, quantification, and prescriptive remediations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 2541–2573, December 2023. doi: 10.18653/v1/2023.emnlp-main.155. URL https://aclanthology.org/2023.emnlp-main.155.

Satoshi Sekine. Development of a question answering system for encyclopedias. *the 9th Annual Meeting of Japanese Association for Natural Language Processing (NLP2003)*, pp. 637–640, March 2003. URL https://www.anlp.jp/proceedings/annual_meeting/2003/pdf_dir/C7-6.pdf.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. Language models are multilingual chain-of-thought reasoners. In *the Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=fR3wGCk-IXp.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *the Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zWqr3MQuNs.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Khyathi Chandu, Jennifer Dumas, Li Lucy, Xinxi Lyu, Ian Magnusson, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Pete Walsh, Hannaneh Hajishirzi, Noah A. Smith, Luke Zettlemoyer, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: An open corpus of 3 trillion tokens for language model pretraining research. Technical report, Allen Institute for AI, 2023. URL https://allenai.github.io/dolma/docs/assets/dolma-datasheet-v0.1.pdf.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, July 2023. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288, 2023.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, July 2023. doi: 10.18653/v1/2023.acl-long.557. URL https://aclanthology.org/2023.acl-long.557.

Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. Poisoning language models during instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35413–35425, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/wan23b.html.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *the Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. How far can camels go? Exploring the state of instruction tuning on open resources. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023b. URL https://openreview.net/forum?id=w4zZNC4ZaV.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, July 2023c. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 4003–4012, May 2020. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.494.

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 *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 483–498, June 2021. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. Large language model unlearning. In *Socially Responsible Language Modelling Research*, 2023. URL https://openreview.net/forum?id=wKe6jE065x.

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 59670–59684, 2024. URL https://proceedings.mlr.press/v235/zhang24ay.html.

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 *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.

A Appendix

A.1 Precise language detection

Following the pipeline of RefinedWeb (Penedo et al., 2023), we built a language detector with a linear discriminator using character n-grams as features. We used multilingual Wikipedia texts as the training data for the linear discriminator, and the feature space was constructed from the training data with character unigrams, bi-grams, and trigrams that satisfy one of the following criteria.

1. Within the top 400,000 occurrences in the training data for all languages.
2. Within the top 400,000 occurrences in the training data of Japanese.
3. Within the top 100,000 occurrences in Chinese training data.
4. Within the top 10,000 occurrences in the training data for each language.

If only Criterion (1) was used, there would be a possibility that features related to Japanese characters would not be obtained sufficiently. Therefore, Criterion (2) aimed to obtain features specific to Japanese. Criterion (3) made it easier to distinguish between Japanese and Chinese, which share Chinese characters. We aimed to stabilize the detection results for texts in languages with insufficient training data using Criterion (4). The total number of distinct features was 821,484.

For training the linear discriminator, we used the dump of Wikipedia CirrusSearch[31] (as of June 12, 2023). In training the language detector, we reduced the amount of training data to 1/20th of the dump because the total amount was too large for training the discriminator. We sampled 100,000 instances in the remaining data for development and evaluation sets. We trained an L2-regularized L2-loss Support Vector Machine on the training data using `LIBLINEAR`[32] 2.46. The regularization coefficient was set to $C = 10$ empirically by the search on the development set. We confirmed that the performance of this classifier was quite good: 0.996 F1 score on the test set and 0.989 F1 score on `whatlang-corpora`[33].

A.2 Rules for removing repetitions

The removal of documents that contain many repetitions aims to filter out documents that do not seem to have useful information and to prevent the behavior of LLMs from repeatedly generating the same words. Specifically, we follow the rules of Rae et al. (2021) and remove a document when the percentage of any of the following exceeds the threshold (shown in parentheses).

1. Number of lines duplicated in other lines / total number of all lines (0.30)
2. Number of paragraphs that are duplicates of other paragraphs / total number of paragraphs (0.30)
3. Number of characters that appear in other lines / total number of all characters (0.20)
4. Number of characters in paragraphs that appear in other paragraphs / total number of characters (0.20)
5. Number of occurrences of the most frequent 2-gram / total number of occurrences of all 2-grams (0.20)
6. Number of occurrences of the most frequent 3-gram / total number of occurrences of all 3-grams (0.18)
7. Number of occurrences of the most frequent 4-gram / total number of occurrences of all 4-grams (0.16)

[31] `https://dumps.wikimedia.org/other/cirrussearch/`

[32] `https://www.csie.ntu.edu.tw/~cjlin/liblinear/`

[33] `https://github.com/whatlang/whatlang-corpora`

8. Total number of occurrences of 5-grams appearing more than once / Total number of occurrences of all 5-grams (0.15)
9. Total number of occurrences of 6-grams appearing more than once / Total number of occurrences of all 6-grams (0.14)
10. Total number of occurrences of 7-grams appearing more than once / Total number of occurrences of all 7-grams (0.13)
11. Total number of occurrences of 8-grams appearing more than once / Total number of occurrences of all 8-grams (0.12)
12. Total number of occurrences of 9-grams appearing more than once / Total number of occurrences of all 9-grams (0.11)
13. Total number of occurrences of 10-grams appearing more than once / Total number of occurrences of all 10-grams (0.10)

A.3 Clearning the text

In order to normalize the punctuation in the corpus to “、” and “。”, we replaced “，” with “、” and “．” with “。”, respectively, based on the following rule.

1. Check the number of occurrences the symbols “、” and “，” appear in the document. If “，” appears more often than “、”, replace “，” with “、”. However, “，” preceding an alphanumeric character is excluded from the replacement with “、”.
2. Check the number of occurrences the symbols “。” and “．” appear in the document. If “．” appears more often than “。”, replace “．” with “。”. However, “。” preceding an alphanumeric character is excluded from the replacement with “．”.

This punctuation normalization process replaced “．” with “。” in 290,318 documents (0.17%) and “，” with “、” in 1,107,319 documents (0.64%).

Although `Trafilatura` used for text extraction removes the navigation and footer text of web pages, we sometimes see footer text that have not be removed by `Trafilatura`. Therefore, expressions such as "Trackback list for this article", "All rights reserved", and "Click" in the last three lines of text were removed from the text if they occupy more than 30% of the text in character. This footer removal process removed footers at the end of 12,617,787 documents (7.3%) of the pages.

A.4 List of LLMs used in the experiments

Base models

- Llama 2 7B: `https://huggingface.co/meta-llama/Llama-2-7b-hf`
- Llama 2 13B: `https://huggingface.co/meta-llama/Llama-2-13b-hf`
- Llama 2 70B: `https://huggingface.co/meta-llama/Llama-2-70b-hf`
- Mistral 7B v0.1: `https://huggingface.co/mistralai/Mistral-7B-v0.1`
- Mixtral 8x7B Instruct v0.1: `https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1`

Other models for comparison

- CyberAgentLM2 7B: `https://huggingface.co/cyberagent/calm2-7b`
- Japanese-StableLM-Base-Beta-7B: `https://huggingface.co/stabilityai/japanese-stablelm-base-beta-7b`
- Rinna Youri 7B: `https://huggingface.co/rinna/youri-7b`
- Qwen 7B: `https://huggingface.co/Qwen/Qwen-7B`

- Rinna Nekomata 7B: `https://huggingface.co/rinna/nekomata-7b`
- Japanese Stable LM Base Gamma 7B:
  `https://huggingface.co/stabilityai/japanese-stablelm-base-gamma-7b`
- Qwen 14B: `https://huggingface.co/Qwen/Qwen-14B`
- KARAKURI LM 70B:
  `https://huggingface.co/karakuri-ai/karakuri-lm-70b-v0.1`
- Japanese-StableLM-Base-Beta-70B:
  `https://huggingface.co/stabilityai/japanese-stablelm-base-beta-70b`
- Qwen 72B: `https://huggingface.co/Qwen/Qwen-72B`