OpenEuroLLM: Open European Family of Large Language Models

Initial Catalogue and Analytics Reports for Existing Training Datasets

Deliverable number: 3.1

Version 1.0

This project has received funding from the Digital Europe Programme under grant agreement No 101195233.

Project Details

Project Acronym:	OpenEuroLLM
Project Full Title:	OpenEuroLLM: Open European Family of Large Language Models
Year of the Call:	2024
Type of Action:	Digital SME Support Actions
Grant Number:	101195233
Project URL:	https://openeurollm.eu

Report Details

Deliverable title	
Lead author:	Stephan Oepen (UiO)
Contributing authors:	Marta Bañón (Prompsit), Aitor Gonzalez-Agirre (BSC), André Maharai Gregussen (UiO) Jindřich Helcl (UiO), Markus Koskela (CSC) Andrey Kutuzov (UiO), Tudor Nicolae Mateiu (Prompsit), Laura Morselli (Cineca), Gema Ramírez Sánchez (Prompsit) Alexander Shvets (BSC)
Internal reviewers:	Trine Platou (ALT-EDIC) Sampo Pyysalo (UTU)
Deliverable number:	3.1
Dissemination level:	Public (PU)
Contractual Delivery Date:	July 31, 2025
Actual Delivery Date:	July 31, 2025
Number of pages:	52

Document History

Version	Date	Changes
1.0	July 31, 2025	Original Submission

Abstract

This report provides an intial description of the OpenEuroLLM Training Data Catalogue, a collectively curated "library" of pre-existing large-scale text collections and associated documentation and statistics. The catalogue is developed and maintained on the LUMI system and automatically mirrored to other AI-enabled EuroHPC systems.

Contents

1. Executive Summary

1.1. Introduction

This deliverable – *D3.1: Initial Catalogue and Analytics Reports for Existing Training Datasets* – corresponds to the verification means for: *WP3: Training Data*, and specifically *T3.1: Training Data Acquisition and Analysis*, in the first six months of the OpenEuroLLM project. WP3 is jointly coordinated by consortium members UiO and Prompsit. Task T3.1 runs over the full project duration. Following the original project proposal, its main goals are to

> *[. . .] collect and categorize existing training datasets for all the targeted languages that will be later curated. Leveraging advanced tools such as HPLT Analytics, thorough analysis (size and lengths statistics, language distribution, presence of noise, etc.) on the gathered datasets will be performed in order to identify the most suitable datasets for further processing. Gathered datasets will be efficiently managed and organized, facilitating easy access and retrieval during subsequent stages of the project. Throughout the project life-cycle, we maintain the initial training dataset catalogue, keeping it up to date. As new training datasets are discovered or created (in Tasks T3.2 and T3.3, or externally), we incorporate them into the catalogue, ensuring its completeness and accuracy. Finally, based on the information gathered from this task, we define a strategy for sourcing new training datasets for targeted languages lacking sufficient or more diverse resources. Close communication with complementary efforts in LANGUAGE-01 [the LLMs4EU project] and other data-gathering projects will avoid unnecessary duplication of efforts. Close cooperation with HPC centers will create immediate availability of training data across Europe.*

This report provides a first status update on the ongoing creation of the OpenEuroLLM Training Data Catalogue (henceforth, just *catalogue*), including the definition of project-internal mechanisms for data curation and management. As of July 2025, an initial version of the catalogue is generally available on the LUMI EuroHPC system, comprising five common pre-training datasets, or some 110 terabytes of compressed data. Ongoing dialogue with consortium members BSC and Cineca makes it appear very likely that the catalogue will be mirrored to the MareNostrum 5 and Leonardo systems in the course of August 2025, and hopefully also the Jupiter system by early fall.

Inclusion of individual datasets in the OpenEuroLLM catalogue, in and of itself, does not constitute a judgment by the project regarding the technical or legal suitability of these resources for LLM development and release. Aspects of regulatory compliance and licensing are briefly discussed in § 2.3 below and will be considered further in separate tasks, e.g. T1.5 (Legal Issues), T3.4 (Training Data Regulatory Compliance), and T4.3 (Dataset Composition).

1.2. Brief Summary of the OpenEuroLLM Project

OpenEuroLLM, with its consortium of 20 European research institutions, companies and EuroHPC centers is building a family of performant, open-source, multilingual, large language foundation models for commercial, industrial and public services. The goal is to lower thresholds for European AI prod-

uct development and refinement, increasing European competitiveness and digital sovereignty. The project also demonstrates the strength of transparency, openness, and community involvement; an Open Strategic Partnership Board has been established to ensure that models, software, data, and evaluation will reflect best practices and be fully open and can be fine-tuned and instruction-tuned for specific industry and public sector needs.

The models will comply with Europe's regulatory framework, ensuring alignment with European values while maintaining technological excellence. OpenEuroLLM leverages support from previous European projects and the experience of the partners and their results, including large repositories of high-quality data and pilot LLMs developed previously. OpenEuroLLM has been awarded the Strategic Technologies for Europe Platform (STEP) seal.

The design and initial implementation of the OpenEuroLLM Training Data Catalogue were coordinated by consortium members UiO and Prompsit, with conceptual and practical contributions from the majority of other consortium members.

2. Initial Catalogue and Analytics Reports

There is a bit of a growth industry in (pre-)training data preparation for LLM development, with new or updated datasets becoming available almost every month, often with central involvement of LLM "heavy-weights" like Allen AI, EleutherAI, Google, HPLT, Hugging Face, Nvidia, etc. The OpenEuroLLM Training Data Catalogue offers navigational help in the dataset landscape, essentially providing a structured 'catalogue' of available resources. The catalogue is curated by the OpenEuroLLM consortium, coordinated by the WP3 leads. Development and community involvement is organized on the GitHub platform (the public pages will also be published via the OpenEuroLLM project web site):

`https://github.com/OpenEuroLLM/training-data-catalogue/blob/main/README.md`

Initially, the catalogue is constructed for internal use in the project, i.e. will put most emphasis on datasets with likely utility for the project and its multilingual perspective. At the same time, it is expected that this overview will become useful to others and can grow into a community-supported resource. The catalogue is accompanied by a curated collection of candidate LLM (pre-)training datasets that are publicly made available (read-only to all users, i.e. not limited to project participants) on multiple AI-enabled EuroHPC systems, currently:

System	Directory Path
LUMI	`/appl/local/openeurollm/training/catalogue/` (production)
Leonardo 5	`/leonardo_work/openeurollm/` (in progress)
MareNostrum 5	`/gpfs/scratch/shared/openeurollm/training/catalogue/` (in progress)

2.1. Background: Motivation & Goals

LLM pre-training from scratch requires trillions of training tokens, and only few initiatives have the technological capacity to create full-scale training datasets themselves. There is an ecosystem of publicly available "de-facto standard" training datasets that most LLM-related R&D needs to build on, going back to English datasets like C4 (the Colossal Clean Crawled Corpus; Raffel et al., 2020) or The Pile (Gao et al., 2020), and more recently FineWeb 1 (Penedo et al., 2025a). Notable large-scale datasets that support a broad range of languages other than English include CulturaX (Nguyen et al., 2024), MADLAD-400 (Kudugunta et al., 2023), and more recently HPLT (de Gibert et al., 2024; Burchell et al., 2025) and FineWeb 2 (Penedo et al., 2025b). Typical LLM training datasets range between a few and tens of terabytes in size each.

LLM R&D, already today, likely accounts for a non-trivial share of activity in the EuroHPC ecosystem, probably with hundreds of distinct initiatives that develop LLMs for various languages or use cases. At present, each LLM project on any of the EuroHPC systems (or other suitable computing environments) needs to perform its own training data management. Seeing as there is a comparatively narrow set of current and actively used training data collections, there is much duplicative effort and redundant storage of (parts of) the same datasets across different projects. The OpenEuroLLM Training Data

Catalogue addresses this wasteful state of affairs by providing a uniform, collectively curated, and well documented collection of candidate LLM training datasets.

The catalogue is built and maintained by expert users – researchers working on LLMs themselves – and automatically mirrored across relevant EuroHPC systems. While primarily motivated for immediate use in OpenEuroLLM, the catalogue is made accessible (read-only) to all users on a specific system, independent of project membership. This design decision implies that the catalogue can only ingest datasets for which the curators can ascertain that general access within the EuroHPC ecosystem is compatible with the terms of use for these resources, a constraint that squares well with the focus on transparency and replicability in OpenEuroLLM – open science.[1] Thus, duplicative effort and storage for common large-scale datasets can be eliminated. The catalogue builds on notions of "discipline-specific self-help" and publicly shared "community directories" that originally were piloted across national systems in Finland and Norway as part of the Nordic Language Processing Laboratory (NLPL; https://nlpl.eu) use case in EOSC-Nordic.[2]

2.2. Implementation: Approach & Technologies

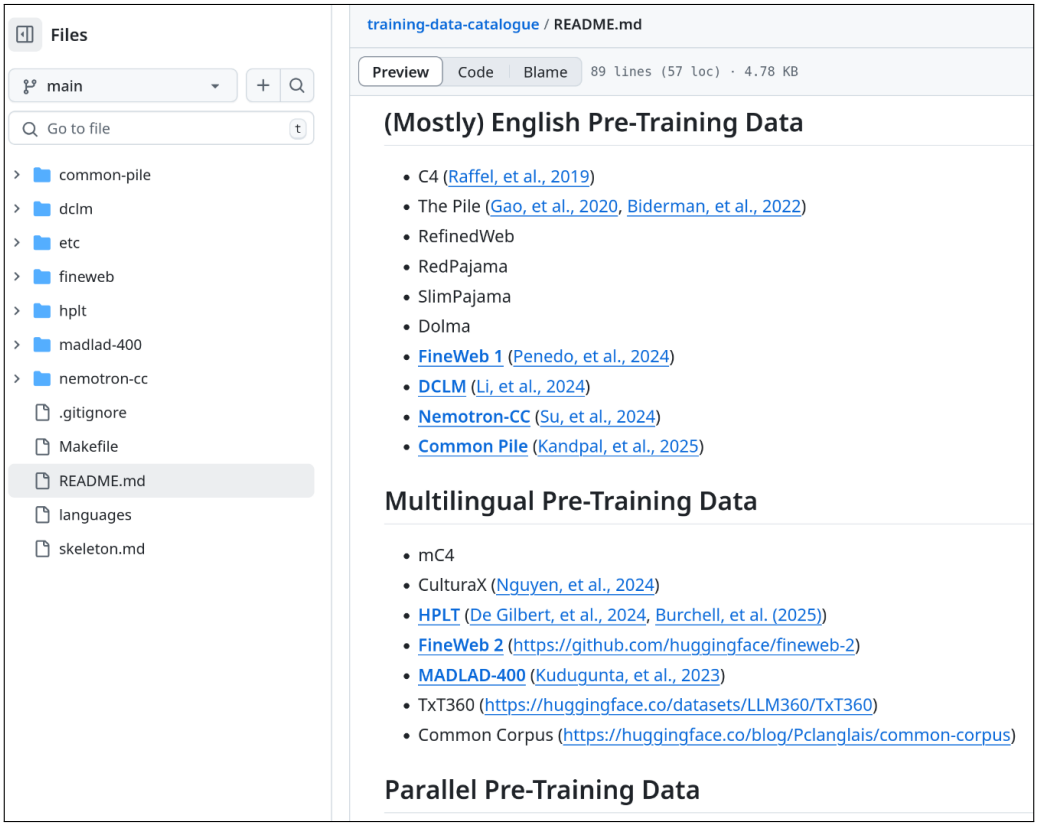


Figure 2.1: Partial screenshot of the top-level catalogue page on GitHub.

[1] In this regard, the existing EuroHPC access policy and mechanisms present an attractive "sandbox" for open, yet not fully unconstrained data sharing. In principle, one can imagine additional layers of access control, for example to limit usage of individual resources to non-commercial R&D only or to establish project-specific access across EuroHPC systems. For the OpenEuroLLM Training Data Catalogue, it is not anticipated that such mechanisms will be required, but they should likely be explored in the implementation of the next generation of AI factories.

[2] https://www.eosc-nordic.eu/kh-usecase/natural-language-processing-lab-for-the-nordics/

The catalogue is a curated resource with mechanisms for collective management and community involvement. Its main content is managed on the GitHub platform, providing basic mechanisms for access control, author attribution, versioning, task and ticket management, community comments, etc. The catalogue uses a tree-like directory structure to represent individual resources and different versions, e.g. `fineweb/1.4.0/`, `fineweb/2.1.0/`, or `hplt/2.0/` for the current versions (as of July 2025) of the FineWeb 1, FineWeb 2, and HPLT datasets, respectively. Each directory contains a descriptive file `README.md`, a set of scripts to automate common tasks like downloading, data validation, and statistics generation, and a set of more in-depth analytics reports (see also § 2.5 and § 2.6 below). All relevant information and scripts for the creation and maintenance of the catalogue are managed on GitHub. The top-level file of the catalogue provides general background information and a table of contents linking to individual catalogue entries. Figure 2.1 shows parts of this top-level navigational page (as of July 2025).[3] Various parts of this deliverable are drawn or adopted from actual catalogue contents on GitHub.

The textual data itself and its metadata, however, are not managed on GitHub, as these usually reside in existing third-party repositories, e.g. the Hugging Face Hub, Common Crawl storage for contributed data, or the HPLT data repository hosted by the Norwegian national research infrastructure. Also, reflecting their size, these resources presuppose dedicated large-scale hosting services. For immediate access on each supported EuroHPC system, the catalogue is instantiated as a shared community directory on individual systems, typically using a logical local path that is independent of a specific and time-limited project or allocation, e.g. `/appl/local/openeurollm/training/catalogue/` on LUMI. This directory is a local `git` clone of the GitHub repository, to guarantee uniformity of the directory structure, metadata, and support scripts.

The community directory is populated through automated download jobs, using scripts versioned in the GitHub repository (see Appendix F for an example). Transferring terabytes of data "over the wire" is not quite a trivial exercise yet, and various hosting sites have specific properties and limitations that the download jobs needs to accommodate, seeking to minimize download time without triggering rate limiting or other throttling mechanisms on the server side. In this regard, the scripts that are part of the catalogue also encode "community wisdom" regarding effective data transfer tools and strategies and could also serve as contrastive connectivity benchmarks. Additionally, each dataset in the catalogue is accompanied by a validation script – seeking to confirm data integrity against whatever metadata or checksums are made available by the original provider; see Appendix G – and scripts to compute basic descriptive statistics (see § 2.5 below) and indices to quantify internal or cross-dataset overlap (see § 2.9 below and Appendix H).

Currently, LUMI serves as the master system for catalogue development. CSC has made available a storage allocation of initially 200 terabytes on the main LUMI filesystem as an in-kind contribution, and UiO has been granted principal investigator status for this allocation for the full OpenEuroLLM duration. Both BSC and Cineca have agreed to mirror this arrangement, i.e. provide matching storage allocations, management privileges for the WP3 leads, and mechanisms for system-wide readability under an "easy-to-find" path. In principle, one could replicate the LUMI catalogue on MareNostrum 5 and Leonardo by executing the same recipes, but the partners will rather work towards fully automated

[3] https://github.com/OpenEuroLLM/training-data-catalogue/blob/main/README.md

mechanism for mirroring across EuroHPC systems, initially a combination of `rsync(1)` and `cron(8)` (as has been used successfully in NLPL before).[4]

The catalogue standardizes on the common JSONlines format, where each document is encoded as a JSON object comprising the document text and available metadata (as defined by each distinct resource), and each JSON object is serialized as a single line, i.e. without internal line breaks. For premium storage efficiency, all files are compressed using the Zstandard (ZSTD) protocol.[5]. Some textual datasets are natively distributed in other formats, notably the FineWeb 1 and FineWeb 2 data in Apache Parquet. In these cases, the data is converted to ZSTD-compressed JSONLines after downloading, to guarantee catalogue-internal uniformity and minimize storage requirements. Compared to its native Parquet format, serializing FineWeb as compressed JSONLines reduces its size by more than a factor of two.

As of July 2025, catalogue construction is in its early stages. Based on consortium-internal deliberation, there is a prioritized list of common datasets to be ingested into the catalogue and initial consensus on the structure and descriptive level of individual entries; see § 2.4 below.

Reflecting the multilingual emphasis of the project, initial catalogue curation has focused on the three most recent broad multilingual datasets, FineWeb 2.1.0, HPLT 2.0, and MADLAD-400 1.0, which each support hundreds of distinct languages. These three datasets form the basis for a preliminary quantitative study of the OpenEuroLLM target languages (see § 2.8 below), which will inform strategic decision making for additional targeted data acquisition in tasks T3.2 and T3.3 of OpenEuroLLM. For ongoing experimentation related to data quality estimation and efficient data use, the catalogue further includes two recent English-only datasets – FineWeb 1.4.0 and Nemotron-CC (Su et al., 2025) – with another two forthcoming, DCLM-baseline (Li et al., 2025) and Common Pile (Kandpal et al., 2025).

Besides large-scale monolingual and multilingual LLM natural language data, the catalogue anticipates entries for parallel textual data (as commonly used in machine translation) and for non-language data, e.g. program code or math and logic datasets. There already is ongoing experimentation with such data in OpenEuroLLM, but the consortium has yet to derive a prioritized list of the most relevant resources.

2.3. Regulatory Compliance

As noted in § 1.1 above (and in the original work package description), the catalogue serves as a "library" of candidate training data for OpenEuroLLM. As such, it does not preclude the selection of specific datasets for the project. Catalogue entries seek to summarize legal aspects of datasets such as their available information about licensing and terms of use. This information, complemented by further investigation and data refinement in other tasks, will help the consortium decide which training data

[4] In a mid-term perspective, the OpenEuroLLM Training Data Catalogue also provides an attractive use case for piloting new cross-system mechanisms for large-scale data scaling, e.g. by use of a tiered, lightweight and read-only caching filesystem like CernVMFS. Typical use of large-scale LLM training data requires one sequential reading pass, to pre-process and tokenize a selection of training data according to project-specific assumptions.

[5] https://datatracker.ietf.org/doc/html/rfc8878

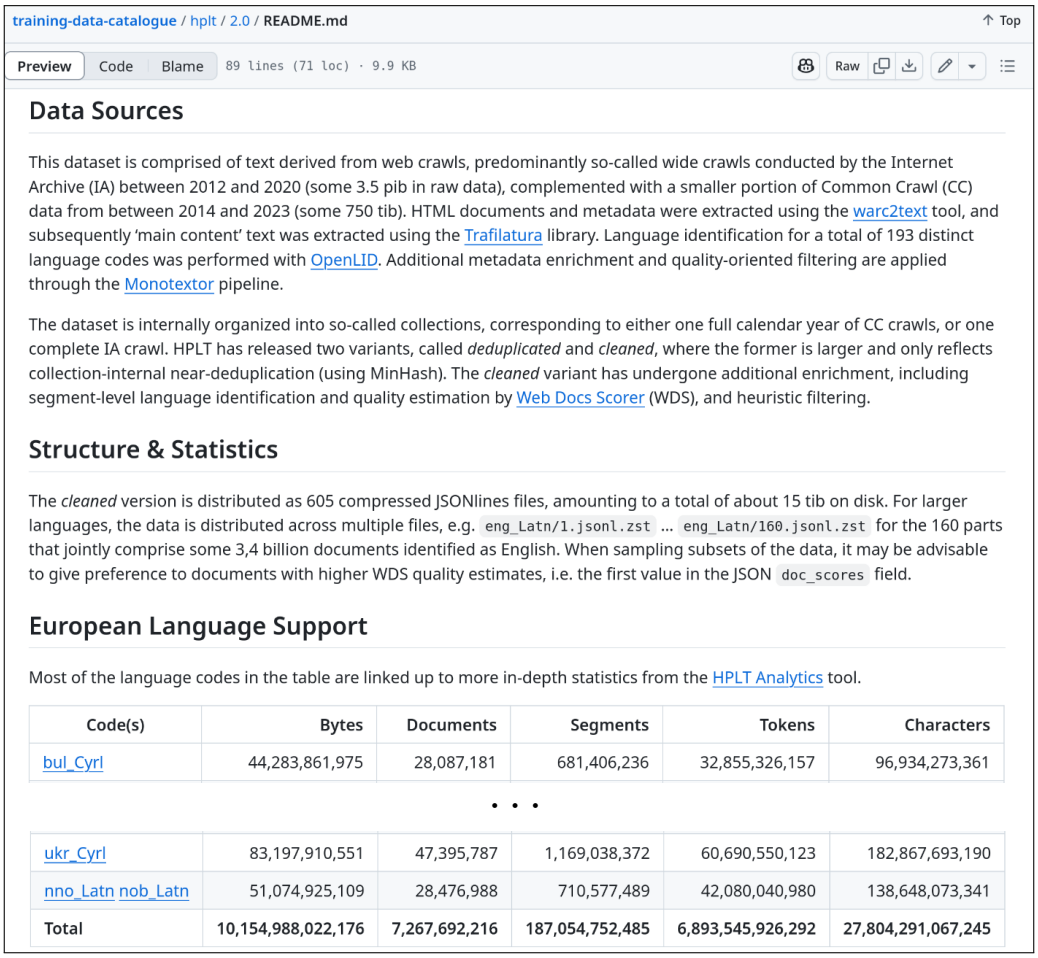
is most suitable for model development and sharing in OpenEuroLLM. Beyond project-internal use, this information will also benefit the broader LLM community.

The catalogue emphasizes "open" datasets with minimal legal uncertainty in using them from the start. To this end, datasets that meet certain criteria have initially been catalogued for more in-depth consideration. Key selection criteria include the following:

- datasets must be generally accessible, for example by public download;
- datasets must provide clear terms of use or licensing information; and
- datasets must not explicitly restrict modification or use for LLM training.

In its final selection and combination of training data, OpenEuroLLM will adhere to applicable EU regulations, such as the provisions in the Copyright Directive, General Data Protection Regulation, AI Act, and compliance measures suggested in the recent General-Purpose AI Code of Practice, notably best practices for rights reservation and opt-out from web crawling.[6]

2.4. Individual Catalogue Entries



training-data-catalogue / hplt / 2.0 / README.md ↑ Top

Preview Code Blame 89 lines (71 loc) · 9.9 KB Raw

Data Sources

This dataset is comprised of text derived from web crawls, predominantly so-called wide crawls conducted by the Internet Archive (IA) between 2012 and 2020 (some 3.5 pib in raw data), complemented with a smaller portion of Common Crawl (CC) data from between 2014 and 2023 (some 750 tib). HTML documents and metadata were extracted using the warc2text tool, and subsequently 'main content' text was extracted using the Trafilatura library. Language identification for a total of 193 distinct language codes was performed with OpenLID. Additional metadata enrichment and quality-oriented filtering are applied through the Monotextor pipeline.

The dataset is internally organized into so-called collections, corresponding to either one full calendar year of CC crawls, or one complete IA crawl. HPLT has released two variants, called *deduplicated* and *cleaned*, where the former is larger and only reflects collection-internal near-deduplication (using MinHash). The *cleaned* variant has undergone additional enrichment, including segment-level language identification and quality estimation by Web Docs Scorer (WDS), and heuristic filtering.

Structure & Statistics

The *cleaned* version is distributed as 605 compressed JSONlines files, amounting to a total of about 15 tib on disk. For larger languages, the data is distributed across multiple files, e.g. `eng_Latn/1.jsonl.zst` … `eng_Latn/160.jsonl.zst` for the 160 parts that jointly comprise some 3,4 billion documents identified as English. When sampling subsets of the data, it may be advisable to give preference to documents with higher WDS quality estimates, i.e. the first value in the JSON `doc_scores` field.

European Language Support

Most of the language codes in the table are linked up to more in-depth statistics from the HPLT Analytics tool.

Code(s)	Bytes	Documents	Segments	Tokens	Characters
bul_Cyrl	44,283,861,975	28,087,181	681,406,236	32,855,326,157	96,934,273,361
• • •					
ukr_Cyrl	83,197,910,551	47,395,787	1,169,038,372	60,690,550,123	182,867,693,190
nno_Latn nob_Latn	51,074,925,109	28,476,988	710,577,489	42,080,040,980	138,648,073,341
Total	**10,154,988,022,176**	**7,267,692,216**	**187,054,752,485**	**6,893,545,926,292**	**27,804,291,067,245**

Figure 2.2: Partial screenshot of the catalogue entry for the HPLT 2.0 dataset.

[6] https://digital-strategy.ec.europa.eu/en/policies/contents-code-gpai

For ease of navigation, each catalogue entry provides a concise, high-level summary in a partly standardized format. The design of these descriptions presents a balancing act between providing relevant information on key properties, on the one hand, while minimizing duplication of technical content from the existing publications or data sheets for each resource, on the other hand. Based on consortium-internal deliberation, a skeletal structure comprising seven sections (as of July 2025) and some editorial conventions have emerged, but this design will likely undergo further refinement as the catalogue evolves, for example to describe standardized samples provided by OpenEuroLLM (see § 2.10 below). The following paragraphs provide short indications of the nature of each of the sections. As an illustrative example, figure 2.2 further shows part of the catalogue entry for the HPLT 2.0 Multilingual Dataset; its full catalogue entry is provided as Appendix A.

Background The creation of very large textual datasets typically reflects the work of larger organizations or collaborative initiatives, and there is an "invisible evolutionary trajectory" behind these developments. This initial section of each catalogue entry seeks to position a specific dataset in the larger landscape of LLM training data collections and provides pointers to the most relevant background publication(s) or web presentations.

Data Sources Common pre-training datasets typically include large components of text derived from web data, e.g. from the Common Crawl, Internet Archive, or other initiatives, and sometimes also include non-web data, e.g. (out-of-copyright) books, government publications, scientific literature, et al. For various sources, text can be derived from different publishing formats, e.g. HTML documents or PDF files extracted from web crawls. It would be hard to devise a formal ontology to fully describe different data sources. Instead, catalogue entries provide a free-text, high-level indication of salient information, e.g. the origin of underlying web crawls and other data sources, original document formats, and such.

Structure & Statistics A collection comprising terabytes of compressed data necessarily will have some internal structure, for example breaking down the data into directories and individual files. Existing datasets differ widely in their internal organization, ranging from a few hundred very large files to tens of thousands of smaller files. Oftentimes this structure reflects relevant dimensions along which the data is organized, for example separation according to document languages, estimated text quality, or provenance of the underlying raw data. This free-text section of each catalogue entry provides high-level guidance on the specific internal structure of each entry.

European Language Support Existing datasets differ widely in their support for and breadth of coverage of different languages, ranging from single-language collections (typically English) to close to two thousand distinct languages. To provide a unifying perspective (and to some degree align the catalogue with the focus of OpenEuroLLM), this section provides tabular summaries of the basic statistics sketched in § 2.5 below, i.e. counts at the level of different structural levels that provide a first indication of the volume of available data in a specific dataset.

Access Information The purpose of the OpenEuroLLM Training Data Catalogue is to liberate LLM developers in the EuroHPC ecosystem from the need to obtain storage allocations and download and organize common training data themselves. This section of each catalogue entry points to the location of the dataset in the filesystem of supported systems (LUMI, as of July 2025, very soon also Leonardo and MareNostrum 5, and later in the fall of 2025 also Jupiter). Additionally, the catalogue provides pointers to the "native" hosting site(s) for each dataset, for example to allow users working outside of EuroHPC to obtain their own local copies.

Terms of Use As discussed briefly in § 2.3, neither the general availability of a specific datasets, nor its inclusion in the OpenEuroLLM catalogue preclude the assessment of legal and regulatory suitability for LLM development and release by the project. This section in each catalogue entry seeks to concisely summarize relevant licensing information or terms of use for a specific dataset, to provide a starting point for catalogue users to make informed decisions about data selection.

Catalogue Curator For each catalogue entry, one team member from the WP3 participants is appointed as the curator of this dataset in the OpenEuroLLM context. Besides taking responsibility for original ingestion and documentation of the data into the catalogue and monitoring it for availability of updated version, it is expected that the curator can serve as a point of contact regarding this catalogue entry both within the consortium and for communication with outside users.

2.5. Analysis: Basic Statistics

To obtain reasonably comparable statistics across different resources, the following metrics are defined:

bytes on-disk size in the catalogue-internal format, compressed JSONLines;

documents number of documents, e.g. web pages, papers, books, or similar;

segments number of paragraph-like units (e.g. `<h1>`, `<p>`, `<li>`, `<pre>` in HTML);

characters total volume in Unicode characters (including whitespace); and

tokens sub-word units according to a common tokenizer (as of July 2025, Gemma3[7]).

Computing these counts over tens of terabytes of compressed data is computationally not quite trivial, in particular as tokenization according to the Gemma3 vocabulary is involved. Upon dataset ingestion into the catalogue, the above statistics are computed as batch jobs on LUMI and stored as "hidden" files in each catalogue entry, viz. as structured summaries for each partition (sub-directory) called `.counts.json`. Among other things, these files allow programmatic generation of summary tables for the GitHub pages (see § 2.4 above) and computation of contrastive statistics like those exemplified in § 2.8 below.

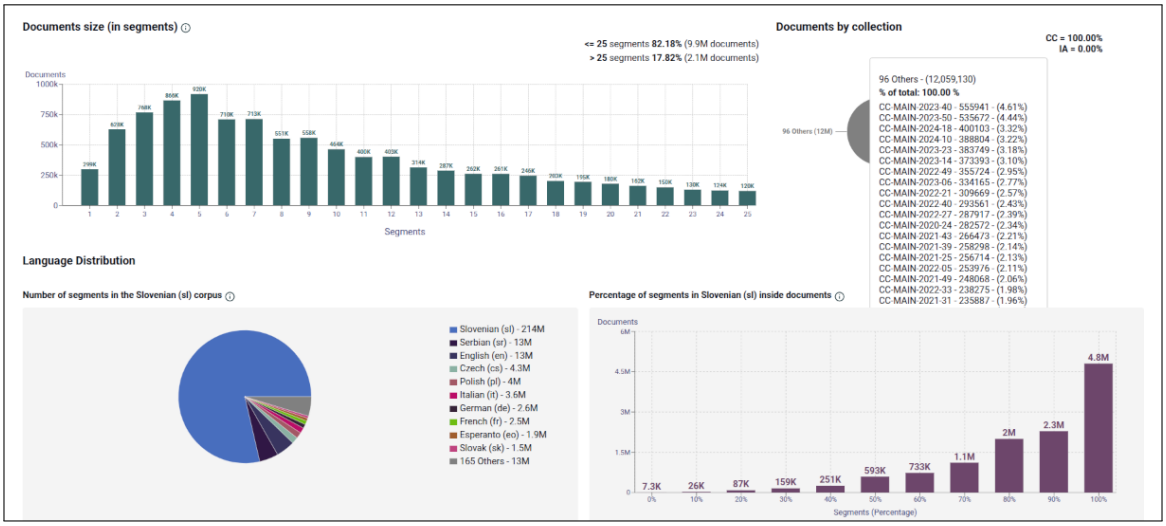


Figure 2.3: Partial screenshot of the HPLT Analytics report for Slovene in FineWeb 2.1.0

2.6. Analysis: HPLT Analytics

For a more comprehensive overview of the data, descriptive analytics reports are also generated with the help of HPLT Analytics. This tool – initially developed within the HPLT project[8] – offers a diverse and fine-grained range of quantitative analytics, catering to both monolingual and parallel datasets.

The current version computes diverse dataset volumes and statistics about the distribution of languages, domains, lengths at document and segment level, noise levels, or quality scores. It also computes common textual types, n-grams and other corpora metrics, gathering insightful information about each dataset. It offers extensive language support and integrates with a web-based dashboard.[9]

The generation of analytics reports is a somewhat heavy-duty batch process. Per-language reports are serialized in YAML syntax, which are versioned on GitHub alongside each catalogue entry. In-browser interactive visualization and PDF generation are supported via the dashboard, and the table providing per-language summary statistics in each catalogue entry (see § 2.4 above) includes direct links to the dashboard, with the specific report preloaded. Figure 2.3 shows a partial screenshot from the interactive visualization of the Slovene subset of FineWeb 2.1.0. The complete analytics report is provided as Appendix B, together with the corresponding report for Slovene in HPLT 2.0 (see Appendix C), and (a "smaller" language) Irish in FineWeb and HPLT (Appendices, D and E, respectively).

For its project duration, OpenEuroLLM will assume the maintenance of the HPLT Analytics tool in close connection to continued refinement of the catalogue; see § 2.10 below for ongoing and planned work.

[7] https://gemma-llm.readthedocs.io/en/latest/colab_tokenizer.html
[8] https://hplt-project.org/
[9] https://analytics.hplt-project.org/

2.7. Target Languages for OpenEuroLLM

Following consortium-internal deliberation during and after the kick-off meeting, the project has established a preliminary list of target languages, taking a strong multilingual and European perspective. In a nutshell, the list emphasizes the official and some co-official languages of EU member states and membership candidates, complemented with the dominant official language of closely associated Norway. In July 2025, the project targets 36 distinct (macro-)languages, with 42 internal variants, e.g. different scripts or written standards. Table 2.1 shows the list of target languages at the time of writing[10], each annotated with specific language–script code combinations observed in the three multilingual datasets analyzed as July 2025 (see § 2.2 above).

For internal consistency, the project standardizes on three-letter language codes from the ISO 639-3 standard[11] and script (or writing system) codes from ISO 15924[12], e.g. `ces_Latn` for Czech in Latin script or `srp_Cyrl` for Serbian in Cyrillic script.

[10] https://github.com/OpenEuroLLM/training-data-catalogue/blob/main/languages
[11] https://en.wikipedia.org/wiki/ISO_639-3
[12] https://en.wikipedia.org/wiki/ISO_15924

Language	(Macro-)Code	Observed Codes
Bulgarian	bul	bul_Cyrl
Czech	ces	ces_Latn
Danish	dan	dan_Latn
German	deu	deu_Latn
Greek	ell	ell_Grek
English	eng	eng_Latn
Estonian	est	est_Latn, ekk_Latn
Finnish	fin	fin_Latn
French	fra	fra_Latn
Irish	gle	gle_Latn
Croatian	hrv	hrv_Latn
Hungarian	hun	hun_Latn
Italian	ita	ita_Latn
Latvian	lav	lav_Latn, ltg_Latn, lvs_Latn
Lithuanian	lit	lit_Latn
Maltese	mlt	mlt_Latn
Dutch	nld	nld_Latn
Polish	pol	pol_Latn
Portuguese	por	por_Latn
Romanian	ron	ron_Latn
Slovak	slk	slk_Latn
Slovene	slv	slv_Latn
Spanish	spa	spa_Latn
Swedish	swe	swe_Latn
Catalan	cat	cat_Latn
Basque	eus	eus_Latn
Galician	glg	glg_Latn
Bosnian	bos	bos_Latn
Georgian	kat	kat_Geor
Macedonian	mkd	mkd_Cyrl
Albanian	sqi	sqi_Latn, als_Latn
Serbian	srp	srp_Cyrl, srp_Latn
Turkish	tur	tur_Latn
Ukrainian	ukr	ukr_Cyrl
Icelandic	isl	isl_Latn
Norwegian	nor	nor_Latn, nno_Latn, nob_Latn

Table 2.1: Preliminary list of target languages for OpenEuroLLM (as of July 2025).[13]

2.8. Preliminary Statistics

Uniform summary statistics and breakdown by individual languages (or other relevant internal structure, in the case of monolingual datasets) in each catalogue entry facilitate cross-dataset comparability

[13] The list of target languages is organized in four groups, official EU languages, co-official languages in in member states, languages from candidate EU members, and other languages with close EU association.

and the definition of derived metrics that may further shed light on the nature and internal structure of specific datasets. To exemplify these notions, Tables 2.3–2.5 provide a breakdown of per-language coverage for the 36 target languages in OpenEuroLLM (as of July 2025; see § 2.7 above) in the three recent multilingual datasets that have been prioritized in catalogue curation to date.[14] For each language, key statistics as defined in § 2.5 have been fetched from the catalogue entries for these three datasets and are presented as consecutive rows in the table, using color-coding to indicate the underlying dataset, viz. counts of *documents*, paragraph-like *segments*[15], Gemma3 *tokens*, and Unicode *characters*. To further indicate the proportional distribution of languages within each dataset, the columns labeled "%" and "%⁻" record what percentage of the total dataset is attributed to this language[16], where the first percentage is calculated over all available data, and the second with the English data excluded. Finally, the column labeled "| |" indicates average document length in tokens.

While MADLAD-400 is a little older, FineWeb 2 and HPLT 2.0 were created independently and at about the same time. There are non-trivial technical differences between the approaches to the creation of the latter two datasets, notably the source of the underlying web archives (Common Crawl vs. Internet Archive), language identification and scope of languages included (1911 vs. 191, respectively), as well as in the specifics of deduplication, data annotation, and filtering. Nevertheless, with the exception of English, raw counts for documents, segments, tokens, and characters often appear broadly comparable, while HPLT exhibits higher average document length for the majority of languages – which may or may not be correlated with text quality. MADLAD-400, on the other hand, is a dataset that has undergone a higher degree of manual per-language inspection and quality control; it is by some believed to exhibit better language identification and text quality, but there are no known systematic experimental studies on these contrasts. In the quantitative perspective of Tables 2.3–2.5, MADLAD-400 shows lower token counts and higher average document length, often substantially so. Several of the languages in the lower tiers of data availability (see below) appear to have comparatively good coverage in MADLAD-400.

For English, Table 2.3 actually uses counts from the catalogue entry for FineWeb 1 (the monolingual English partition), which was originally created as a separate dataset in its own right, arguably focusing more on volume than on quality data filtering, and only applied deduplication within each of its 104 crawls (Penedo et al., 2025a). Therefore, the proportion of English data for FineWeb appears substantially higher than for HPLT and MADLAD-400 – at 88% compared to 57% and 49%, respectively – and the individual proportions for other languages in this perspective are correspondingly lower in FineWeb. When adjusting for the stark imbalance in English data across the three datasets, i.e. the proportions relative to the sum of all other languages in the column labeled "%⁻", the relative per-language shares across datasets also become more similar, which arguably leaves the differences in average document lengths as the most noteworthy observation.

Seeing as there is substantial overlap between the three datasets, a conservative measure of available

[14] Each of these dataset includes additional languages, which are not taken into consideration in these statistics. Thus, the total counts in Tables 2.3–2.5 only reflect the subsets of each dataset corresponding to the OpenEuroLLM target languages, but these do include the languages that account for the vast majority of the available data.

[15] From this contrastive view, it emerges that MADLAD-400 does not provide document-internal segment structure.

[16] Based, of course, on the automated document-level language identification performed by the original dataset creators, which introduces some limits to cross-dataset comparability.

textual data per language is to use the maximum token count observed (rather than summing across the datasets). Using this approach, one can arrange the target languages for OpenEuroLLM in tiers according to orders of magnitude of readily available data:

Tokens	Languages
$\geq 1,000,000,000,000$ (one trillion)	eng
$\geq 100,000,000,000$ (hundred billion)	deu, fra, ell, ita, nld, pol, por, spa
$\geq 10,000,000,000$ (ten billion)	bos, bul, cat, ces, dan, est, fin, hrv, hun, lav, lit, nor, ron, slk, slv, swe, tur, ukr
$\geq 1,000,000,000$ (one billion)	eus, glg, isl, kat, mkd, sqi, srp
$< 1,000,000,000$ (one billion)	gle, mlt

Table 2.2: Tiers of available textual data for OpenEuroLLM target languages.

Based on these preliminary observations, the languages in the lower tiers of this hierarchy will require special attention in tasks T3.2 and T3.3 of the work package, i.e. targeted acquisition of additional training data.

Language	Documents	Segments	Tokens	%	%⁻	||	Characters
	13,998,330	277,731,853	15,407,308,842	0.08	0.67	1100.7	46,817,643,742
bul	28,087,181	681,406,236	32,855,326,157	0.48	1.10	1169.8	96,934,273,361
	12,755,329	12,755,329	19,489,755,290	0.57	1.12	1528.0	57,828,623,862
	17,122,682	201,272,389	12,249,745,925	0.06	0.53	715.4	41,984,711,079
cat	18,553,883	383,335,831	18,116,292,562	0.26	0.61	976.4	60,186,591,495
	9,477,390	9,477,390	10,279,279,397	0.30	0.59	1084.6	34,572,051,501
	66,095,955	1,306,374,161	67,104,577,548	0.35	2.91	1015.3	200,038,397,804
ces	75,288,021	1,926,503,033	95,363,069,335	1.38	3.20	1266.6	273,936,688,894
	38,254,671	38,254,671	50,851,804,344	1.48	2.91	1329.3	147,891,897,647
	45,419,600	1,031,784,537	46,560,059,625	0.24	2.02	1025.1	158,243,598,985
dan	33,841,408	873,022,625	41,156,519,209	0.60	1.38	1216.2	133,380,682,616
	17,865,888	17,865,888	24,933,594,934	0.73	1.43	1395.6	83,083,948,094
	474,465,528	8,332,457,880	385,295,978,051	1.99	16.68	812.1	1,561,407,716,959
deu	482,053,407	11,127,774,286	449,431,582,918	6.52	15.09	932.3	1,782,129,825,333
	225,111,495	225,111,495	252,228,862,830	7.36	14.45	1120.5	1,009,466,023,318
	47,454,009	664,498,640	50,957,983,721	0.26	2.21	1073.8	133,425,064,206
ell	70,328,890	1,849,481,662	115,599,058,101	1.68	3.88	1643.7	283,534,611,644
	20,932,239	20,932,239	31,798,629,582	0.93	1.82	1519.1	80,937,608,560
	24,505,935,751	390,005,708,780	17,039,468,180,748	88.06		695.3	76,522,410,156,728
eng	4,388,525,961	116,521,950,325	3,915,588,774,525	56.80		892.2	17,083,161,859,947
	1,528,918,474	1,528,918,474	1,681,401,607,318	49.06		1099.7	7,442,380,691,991
	10,244,373	271,180,584	14,263,514,314	0.07	0.62	1392.3	42,646,095,690
est	8,449,320	264,422,814	12,324,211,253	0.18	0.41	1458.6	36,018,221,232
	5,542,933	5,542,933	9,721,783,976	0.28	0.56	1753.9	28,682,517,462
	1,585,226	23,161,056	1,508,546,208	0.01	0.07	951.6	4,645,312,202
eus	1,974,218	37,621,611	2,034,478,450	0.03	0.07	1030.5	6,052,165,410
	1,155,671	1,155,671	1,454,331,118	0.04	0.08	1258.4	4,318,534,844
	36,741,214	894,429,775	48,679,163,495	0.25	2.11	1324.9	148,737,305,693
fin	34,815,601	976,622,086	53,580,820,308	0.78	1.80	1539.0	155,678,802,052
	20,433,664	20,433,664	34,201,262,623	1.00	1.96	1673.8	101,061,286,660
	343,504,739	6,379,785,382	278,951,494,784	1.44	12.08	812.1	1,123,447,206,256
fra	401,831,660	10,557,148,321	379,038,708,184	5.50	12.73	943.3	1,457,428,851,611
	216,945,532	216,945,532	269,040,813,325	7.85	15.41	1240.1	1,035,390,241,419
	653,723	13,608,630	804,760,389	0.00	0.03	1231.0	2,218,321,271
gle	490,787	10,993,158	643,453,119	0.01	0.02	1311.1	1,749,350,336
	285,999	285,999	519,239,773	0.02	0.03	1815.5	1,409,195,138
	2,526,106	32,619,555	1,756,331,487	0.01	0.08	695.3	6,779,187,629
glg	3,020,164	61,177,888	2,736,491,963	0.04	0.09	906.1	10,108,660,186
	1,253,170	1,253,170	1,258,531,316	0.04	0.07	1004.3	4,760,109,761
	6,205,072	221,619,431	11,924,928,106	0.06	0.52	1921.8	37,523,929,825
hrv	12,303,820	297,132,744	15,377,672,465	0.22	0.52	1249.8	47,995,473,960
	2,841,400	2,841,400	3,065,365,777	0.09	0.18	1078.8	9,598,383,639
	49,970,765	1,100,271,787	66,679,144,069	0.34	2.89	1334.4	196,432,800,732
hun	51,870,492	1,418,772,876	79,082,122,145	1.15	2.66	1524.6	225,200,264,565
	29,677,075	29,677,075	46,370,503,350	1.35	2.66	1562.5	134,919,754,643

Table 2.3: Contrastive statistics for Fineweb 2.1.0, HPLT 2.0, and MADLAD-400 1.0).

Language	Documents	Segments	Tokens	%	%⁻	||	Characters
	222,930,894	3,790,928,376	184,070,864,906	0.95	7.97	825.7	732,718,297,963
ita	221,752,424	5,127,292,899	213,754,351,761	3.10	7.18	963.9	820,602,938,696
	126,406,256	126,406,256	141,033,815,308	4.12	8.08	1115.7	553,099,788,690
	8,029,516	194,617,747	12,432,815,934	0.06	0.54	1548.4	32,823,495,101
lav	6,780,843	173,958,974	9,777,313,720	0.14	0.33	1441.9	25,209,419,142
	5,007,982	5,007,982	9,200,513,017	0.27	0.53	1837.2	23,908,350,853
	13,499,659	342,421,580	20,117,860,541	0.10	0.87	1490.2	57,192,092,882
lit	13,338,275	322,156,374	17,999,481,637	0.26	0.60	1349.5	50,393,738,585
	8,748,025	8,748,025	14,590,832,555	0.43	0.84	1667.9	41,272,592,571
	494,383	9,660,602	625,839,053	0.00	0.03	1265.9	1,589,701,021
mlt	367,265	8,675,475	570,825,363	0.01	0.02	1554.3	1,441,648,250
	265,388	265,388	539,311,594	0.02	0.03	2032.2	1,343,837,424
	147,334,553	2,464,203,353	111,879,668,657	0.58	4.84	759.4	425,299,182,167
nld	138,651,084	3,074,592,386	122,628,893,009	1.78	4.12	884.4	451,077,252,328
	86,594,116	86,594,116	91,087,694,983	2.66	5.22	1051.9	334,538,938,666
	145,593,900	2,568,048,983	127,012,109,404	0.66	5.50	872.4	421,223,906,202
pol	175,410,669	4,460,832,917	196,052,655,218	2.84	6.58	1117.7	631,594,269,186
	90,908,786	90,908,786	111,112,599,813	3.24	6.37	1222.2	356,352,714,277
	199,308,976	3,382,337,623	146,332,360,777	0.76	6.34	734.2	591,736,617,740
por	237,812,825	6,124,611,786	233,189,157,063	3.38	7.83	980.6	896,547,444,407
	124,207,090	124,207,090	127,079,774,190	3.71	7.28	1023.1	499,798,407,634
	58,349,454	1,075,272,363	57,003,579,440	0.29	2.47	976.9	192,237,271,824
ron	65,876,383	1,696,970,479	76,264,228,246	1.11	2.56	1157.7	250,658,132,448
	35,397,563	35,397,563	44,698,078,125	1.30	2.56	1262.7	148,161,405,070
	30,024,078	599,150,608	29,150,677,810	0.15	1.26	970.9	84,625,527,586
slk	21,827,259	494,278,579	24,504,432,765	0.36	0.82	1122.7	70,372,196,449
	11,857,945	11,857,945	15,828,510,566	0.46	0.91	1334.8	45,655,828,920
	12,091,577	275,687,954	14,479,201,093	0.07	0.63	1197.5	43,902,424,340
slv	10,277,173	238,644,943	11,867,536,246	0.17	0.40	1154.7	35,258,183,993
	6,310,419	6,310,419	10,247,956,241	0.30	0.59	1624.0	30,539,910,053
	427,278,659	6,798,523,945	319,086,364,464	1.65	13.82	746.8	1,374,263,502,044
spa	503,073,098	12,121,752,157	471,218,993,500	6.84	15.82	936.7	1,953,862,248,952
	250,906,994	250,906,994	254,750,449,059	7.43	14.59	1015.3	1,063,143,461,628
	59,509,998	1,140,171,555	60,273,268,862	0.31	2.61	1012.8	210,108,849,997
swe	66,812,562	1,754,677,064	75,784,600,156	1.10	2.54	1134.3	251,109,959,822
	35,153,050	35,153,050	46,075,028,237	1.34	2.64	1310.7	153,667,702,685
	21,292,528	308,427,409	16,348,958,893	0.08	0.71	767.8	51,314,524,140
bos	14,613,088	268,156,648	14,828,824,339	0.22	0.50	1014.8	46,070,953,520
	6,226	6,226	59,234,247	0.00	0.00	9514.0	88,693,363
	3,047,015	55,668,501	3,912,928,195	0.02	0.17	1284.2	10,134,824,359
isl	2,840,735	69,643,257	3,835,365,590	0.06	0.13	1350.1	9,593,246,968
	1,560,913	1,560,913	2,591,002,405	0.08	0.15	1659.9	6,356,784,021
	3,738,257	59,737,599	4,527,566,114	0.02	0.20	1211.1	10,357,741,132
kat	3,335,164	63,722,098	4,538,769,891	0.07	0.15	1360.9	10,155,612,392
	936,497	936,497	1,721,264,158	0.05	0.10	1838.0	3,833,662,238

Table 2.4: Contrastive statistics for Fineweb 2.1.0, HPLT 2.0, and MADLAD-400 1.0).

Language	Documents	Segments	Tokens	%	%⁻	||	Characters
mkd	4,192,851	48,341,056	3,218,958,253	0.02	0.14	767.7	9,062,537,159
	3,565,647	57,008,331	3,406,651,991	0.05	0.11	955.4	9,439,624,767
	1,358,293	1,358,293	1,602,877,239	0.05	0.09	1180.1	4,451,886,420
sqi	8,661,004	101,210,254	6,592,518,537	0.03	0.29	761.2	18,243,668,440
	5,385,262	95,101,980	5,892,424,412	0.09	0.20	1094.2	16,095,653,237
	3,622,957	3,622,957	4,615,976,832	0.13	0.26	1274.1	12,667,553,241
srp	4,778,149	102,640,053	6,678,431,907	0.03	0.29	1397.7	18,352,667,670
	4,123,458	93,809,457	6,106,504,834	0.09	0.21	1480.9	16,156,879,041
	2,010,607	2,010,607	4,104,764,692	0.12	0.24	2041.6	10,957,011,175
tur	95,165,275	1,608,161,042	80,015,693,931	0.41	3.46	840.8	279,433,007,976
	84,541,414	1,941,885,324	85,625,744,754	1.24	2.88	1012.8	283,639,575,889
	54,327,085	54,327,085	55,254,256,268	1.61	3.17	1017.1	191,482,314,505
ukr	53,142,014	957,094,926	49,892,422,604	0.26	2.16	938.9	151,652,717,758
	47,395,787	1,169,038,372	60,690,550,123	0.88	2.04	1280.5	182,867,693,190
	24,968,305	24,968,305	31,677,007,258	0.92	1.81	1268.7	95,173,965,329
nor	39,414,836	917,158,247	53,533,100,559	0.28	2.32	1358.2	184,207,546,884
	28,476,988	710,577,489	42,080,040,980	0.61	1.41	1477.7	138,648,073,341
	14,864,710	14,864,710	22,418,792,894	0.65	1.28	1508.2	74,784,171,188
Total	7,267,692,216	437,555,968,216	19,348,796,907,246	100	100	711.9	85,127,237,553,186
	7,267,692,216	187,054,752,485	6,893,545,926,292	100	100	948.5	27,804,291,067,245
	7,267,692,216	3,016,870,137	3,426,905,104,634	100	100	1135.9	13,827,579,848,490

Table 2.5: Contrastive statistics for Fineweb 2.1.0, HPLT 2.0, and MADLAD-400 1.0).

2.9. Overlap: Sample Analysis

Bulk LLM training data inevitably has a large core of web data, where most datasets build (near-) exclusively on raw web archives from the Common Crawl Foundation. Among the current catalogue entries, HPLT is the exception to this rule, but it too includes a non-trivial proportion of text derived from Common Crawl archives. Also, even wholly independent crawls – starting from different seeds or datacenter regions, and using distinct strategies – will likely exhibit some degree of overlap. Furthermore, datasets differ in their approach to duplication, both in (a) what is considered a (near-) duplicate, (b) whether deduplication is applied at the document or segment levels, and (c) global vs. per-crawl deduplication, i.e. the context window for duplicate discovery.

To estimate to what degree different datasets actually provide overlapping data, internally and in comparison to another dataset, the catalogue provides tooling for the creation and comparison of *indices* composed of three indicative properties of individual documents: (a) internet domains, (b) full web addresses (URLs), and (c) what is called normalized signatures. The latter of these notions is intended as a computationally cheap proxy for document-level, semi-fuzzy duplicate detection, where each document is normalized by removing non-word characters and downcasing, and the resulting string indexed by its MD5 checksum. It is expected that these perspectives also will facilitate "mixing & matching" across datasets (see § 2.10 below) to support experimentation with different combinations of data. Parallel to the basic statistics of § 2.5, indices are serialized directly into each catalogue (sub-) directory, as "hidden" files `.domains.zst`, `.urls.zst`, and `.signatures.zst`. For interoperability with standard Un*x tools, these files use a tabulator-separated format, where each line records a triple comprised of (a) a key (a unique domain, URL, or signature), (b) the corresponding count, and (c) a JSON object recording for each underlying data file the line indices where this key occurred. The files are sorted by keys, i.e. the first field of each line.

The catalogue includes emerging code to intersect two indices and summarize the results. For a given type of key, say URLs, and a pair of indices $\mathcal{I}$ and $\mathcal{J}$, this process identifies the keys that occur in both $\mathcal{I}$ and $\mathcal{J}$ and sums their frequency of occurrence. The overlap $\cap(\cdot)$ of $\mathcal{I}$ with regard to $\mathcal{J}$ is defined as the ratio between the frequency sum in the intersection and the total frequency sum for $\mathcal{I}$. Assume

$$\mathcal{I} = \{A, A, B, C, D\}$$
$$\mathcal{J} = \{A, B, B, B\}$$

As $\mathcal{I}$ has two entries in common with $\mathcal{J}$ and total cardinality (frequency sum) five, $\cap(\mathcal{I}, \mathcal{J}) = \frac{2}{5} = 0.4$. Unlike the closely related Jaccard index, if generalized to multi-sets, the definition of $\cap(\cdot)$ of $\mathcal{I}$ is not symmetric with regard to $\mathcal{I}$ and $\mathcal{J}$: $\cap(\mathcal{J}, \mathcal{I}) = \frac{2}{4} = 0.5 \neq \cap(\mathcal{I}, \mathcal{J})$. This reflects the intuitive observation that a larger fraction of (the smaller multi-set) $\mathcal{J}$ overlaps with $\mathcal{I}$ than vice versa.

Finally, a notion of "self-overlap" can also be applied meaningfully when comparing a multi-set to itself, viz. to quantify the degree of dataset-internal duplication. This can be a relevant measure because data repetition is the inverse of internal variation and, therefore, can negatively impact LLM training (Lee et al., 2022). In the above example, $\mathcal{I}$ intuitively exhibits greater internal variation than $\mathcal{J}$, and indeed $\cap(\mathcal{I}, \mathcal{I}) = 0.2$, whereas $\cap(\mathcal{J}, \mathcal{J}) = 0.5$. Table 2.6 exemplifies these notions for the Norwegian subset

	FineWeb	HPLT	MADLAD
FineWeb	98.7	46.6	–
HPLT	64.5	98.1	–
MADLAD	–	–	–

(a) $\cap(\cdot)$ internet domains;

	FineWeb	HPLT	MADLAD
FineWeb	8.4	24.7	–
HPLT	34.2	20.3	–
MADLAD	–	–	–

(b) $\cap(\cdot)$ web addresses (URLs);

	FineWeb	HPLT	MADLAD
FineWeb	0.0	14.5	0.0
HPLT	20.1	20.5	0.0
MADLAD	0.0	0.0	0.0

(c) $\cap(\cdot)$ normalized signatures.

Table 2.6: Overlaps in Norwegian documents among and within three datasets, for different metrics.

(the penultimate row in Table 2.5) of the three multilingual datasets that OpenEuroLLM initially has prioritized (as of July 2025), FineWeb 2.1.0, HPLT 2.0, and MADLAD-400 1.0.

These statistics suggest that FineWeb and HPLT build on web content from many of the same domains, though the overlap in specific URLs and document signatures is substantially lower; the two datasets appear somewhat complementary for Norwegian. Comparison to MADLAD-400 is difficult however, seeing as the dataset does not record URLs from which textual content was extracted, and also taking into account that it used a different approach to text extraction from the original HTML documents. It appears likely that the basic normalization applied in the document signatures is insufficient to detect near-equivalent documents, i.e. texts that only differ in, for example, the inclusion or absence of "boilerplate" elements. Future work on tooling for near-duplicate detection (see § 2.10 below) will explore more robust signatures like MinHash or Bloom filters, as discussed by Lee et al. (2022) and used internally by the original dataset creators. Finally, in terms of dataset-internal "self-overlap", the different approaches of FineWeb and MADLAD-400, on the one hand, and HPLT, on the other hand, are visible in the diagonals of Table 2.6. Following the experience report of Penedo et al. (2025a), HPLT applied near-deduplication at the level of internal partitions called *collections*, essentially one Internet Archive crawl or one calendar year of Common Crawl archives, whereas the other two datasets applied global deduplication.

For the English-only FineWeb 1.4.0, Table 2.7 quantifies pairwise overlap among a sample of internal partions, ten Common Crawl snapshots ranging between 2013 and 2024. Using the pre-computed indices, in this case over normalized document signatures, these statistics can be derived in about half an hour on a single "small" LUMI compute node. Similar to HPLT 2.0, the English version of FineWeb applied deduplication only within each crawl. The table shows substantial variation over time, where some pairs of crawls exhibit massive overlap – e.g. 50.6% for CC-MAIN-2016-26 and CC-MAIN-2015-14 – whereas other pairs appear almost fully complementary – e.g. CC-MAIN-2013-20 and CC-MAIN-2024-51, with only 0.4% overlap. Overall, pairwise overlap diminishes with larger temporal distance between crawls, and average overlap levels are higher for older crawl pairs than for the more recent ones. Complementing the study of this sub-set of crawls, the full FineWeb 1.4.0 exhibits 22.4% self-overlap, higher than an average of the pairwise comparison of the sample in Table 2.7 might seem to suggest. This reflects higher temporal density of crawls in the full dataset, with a total of 104 snapshots in the interval between CC-MAIN-2013-20 and CC-MAIN-2024-51.

	CC-MAIN-2013-20	CC-MAIN-2015-14	CC-MAIN-2016-26	CC-MAIN-2017-30	CC-MAIN-2018-26	CC-MAIN-2019-26	CC-MAIN-2020-29	CC-MAIN-2021-43	CC-MAIN-2023-40	CC-MAIN-2024-51
CC-MAIN-2013-20	–	22.6	15.3	6.7	3.8	1.3	1.2	1.3	1.0	0.4
CC-MAIN-2015-14	26.9	–	36.8	12.8	6.5	2.0	1.7	1.8	1.3	0.5
CC-MAIN-2016-26	25.0	50.6	–	19.2	9.3	2.7	2.2	2.2	1.7	0.7
CC-MAIN-2017-30	6.0	9.6	10.5	–	23.3	7.7	6.0	5.4	3.0	1.2
CC-MAIN-2018-26	3.3	4.7	4.9	22.5	–	15.4	9.2	7.4	4.1	1.6
CC-MAIN-2019-26	1.3	1.7	1.6	8.6	17.7	–	20.0	12.8	6.8	2.5
CC-MAIN-2020-29	1.0	1.2	1.1	5.6	9.0	16.9	–	19.2	9.5	3.3
CC-MAIN-2021-43	1.0	1.1	1.0	4.3	6.1	9.2	16.3	–	13.4	4.5
CC-MAIN-2023-40	0.6	0.7	0.7	2.2	3.13	4.5	7.4	12.3	–	8.2
CC-MAIN-2024-51	0.5	0.5	0.5	1.6	2.2	3.0	4.8	7.6	15.1	–

Table 2.7: Overlap between a sample of internal partitions in English FineWeb 1.4.0.

2.10. Outlook: Management & Tooling

The initial catalogue on LUMI is in production as of July 2025, and mirroring to the Leonardo and MareNostrum 5 systems is expected to fall into place in a matter of a few weeks. Active use and public access to the catalogue calls for well-defined procedures – by-laws and governance structures – for (a) selection of additional datasets to ingest and (b) deprecation and removal of "out-of-date" resources over time. For example, Hugging Face regularly provides updates to the FineWeb datasets, e.g. versions 1.0.0, 1.1.0, ..., 1.4.0 between May 2024 and July 2025 for the original English partition. Each full version requires about 25 terabytes of storage. It seems plausible to expect that new experimentation should start from the latest version of each dataset in the catalogue, but some users might be in the middle of a series of experiments, based on an earlier version, and would be inconvenienced by abrupt loss of access to this data. Thus, we anticipate that the catalogue can provide multiple versions of a dataset, where superseded versions will be flagged as deprecated and removed after a grace period of, for example, three to six months.

In ongoing technical work, the HPLT and OpenEuroLLM teams collaborate to scale up HPLT Analytics for automated processing of datasets in the 15-trillion-token scale of FineWeb 1.4.0 on LUMI, so that generation of analytics reports becomes a fully integral part of the ingestion of new datasets. Additionally, building on the indices and overlap statistics of § 2.9, the consortium will develop tooling to support data selection and combination from the catalogue in task T3.5 (Data Configuration and Packaging). This work builds, among others, on the experiences from the HPLT and TrustLLM consortia and will allow flexible "mixing & matching" across datasets, for example selection of data by arbitrary metadata properties and application of diverse de-duplication strategies, e.g. at the document

or segment levels and offering different degrees of fuzziness.

To support this work, one extension of the skeletal structure for catalogue entries (see § 2.4 above) is already being prepared, viz. an additional section to summarize the extent and nature of available annotations and metdata in each dataset, like for example the document quality estimates and register labels in HPLT. Another short-term extension will be the creation of standard samples of different sizes for all datasets. While some datasets make available pre-defined samples (sub-sets of e.g. one or ten billion tokens) of the data, for example to faciliate faster exploration or human inspection, such pre-defined sub-sets follow different design critera, and other datasets do not provide such samples at all. Reflecting the collective "wisdom" of the consortium and the specific use cases in OpenEuroLLM, the catalogue will be augmented with pre-computed data samples in a uniform and systematic approach, as well as with additional tooling for users to produce custom samples.

Finally, in a mid-term perspective, several of the activities in work packages WP4 (Training) and WP5 (Evaluation) need to be able to search the pre-training data at the level of individual documents, segments, or arbitrary sub-sequences of tokens. At the scale of tens of trillions of tokens, efficient full-text search is both algorithmically and technologically a non-trivial task and will likely require assistance from the HPC partners. The consortium will seek to scale up preliminary work from HPLT (off-line search based on MinHash signatures) and explore the recent infrastructure of Xu et al. (2025) for more immediate, interactive search.

A. Example Catalogue Entry: HPLT Multilingual Datasets 2.0

HPLT Monolingual Datasets (Version 2.0; of September 2024)

Background

The HPLT Monolingual Datasets 2.0 aim to provide large volumes of high-quality running text with strong multilingual emphasis. The data construction and preliminary experimental results are described by Burchell, et al., 2025, to appear in the Proceedings of the Annual Conference of the Association for Computational Linguistics. Additional details and download instructions are available from the HPLT download site.

Data Sources

This dataset is comprised of text derived from web crawls, predominantly so-called wide crawls conducted by the Internet Archive (IA) between 2012 and 2020 (some 3.5 pib in raw data), complemented with a smaller portion of Common Crawl (CC) data from between 2014 and 2023 (some 750 tib). HTML documents and metadata were extracted using the warc2text tool, and subsequently 'main content' text was extracted using the Trafilatura library. Language identification for a total of 193 distinct language codes was performed with OpenLID. Additional metadata enrichment and quality-oriented filtering are applied through the Monotextor pipeline.

The dataset is internally organized into so-called collections, corresponding to either one full calendar year of CC crawls, or one complete IA crawl. HPLT has released two variants, called *deduplicated* and *cleaned*, where the former is larger and only reflects collection-internal near-deduplication (using MinHash). The *cleaned* variant has undergone additional enrichment, including segment-level language identification and quality estimation by Web Docs Scorer (WDS), and heuristic filtering.

Structure & Statistics

The *cleaned* version is distributed as 605 compressed JSONlines files, amounting to a total of about 15 tib on disk. For larger languages, the data is distributed across multiple files, e.g. `eng_Latn/1.jsonl.zst` ... `eng_Latn/160.jsonl.zst` for the 160 parts that jointly comprise some 3,4 billion documents identified as English. When sampling subsets of the data, it may be advisable to give preference to documents with higher WDS quality estimates, i.e. the first value in the JSON `doc_scores` field.

European Language Support

Most of the language codes in the table are linked up to more in-depth statistics from the HPLT Analytics tool.

Code(s)	Bytes	Documents	Segments	Tokens	Characters
bul_Cyrl	44,283,861,975	28,087,181	681,406,236	32,855,326,157	96,934,273,361
ces_Latn	109,711,940,916	75,288,021	1,926,503,033	95,363,069,335	273,936,688,894
dan_Latn	46,874,204,383	33,841,408	873,022,625	41,156,519,209	133,380,682,616
deu_Latn	643,563,226,429	482,053,407	11,127,774,286	449,431,582,918	1,782,129,825,333
ell_Grek	125,752,059,355	70,328,890	1,849,481,662	115,599,058,101	283,534,611,644
eng_Latn	6,199,414,043,792	4,388,525,961	116,521,950,325	3,915,588,774,525	17,083,161,859,947
est_Latn	13,143,473,236	8,449,320	264,422,814	12,324,211,253	36,018,221,232
fin_Latn	55,164,578,152	34,815,601	976,622,086	53,580,820,308	155,678,802,052
fra_Latn	528,153,012,485	401,831,660	10,557,148,321	379,038,708,184	1,457,428,851,611
gle_Latn	608,544,067	490,787	10,993,158	643,453,119	1,749,350,336
hrv_Latn	18,455,135,510	12,303,820	297,132,744	15,377,672,465	47,995,473,960
hun_Latn	84,104,083,079	51,870,492	1,418,772,876	79,082,122,145	225,200,264,565
ita_Latn	298,427,404,410	221,752,424	5,127,292,899	213,754,351,761	820,602,938,696
lvs_Latn ltg_Latn	9,240,310,207	6,780,843	173,958,974	9,777,313,720	25,209,419,142
lit_Latn	18,792,388,046	13,338,275	322,156,374	17,999,481,637	50,393,738,585
mlt_Latn	473,820,795	367,265	8,675,475	570,825,363	1,441,648,250
nld_Latn	163,348,254,430	138,651,084	3,074,592,386	122,628,893,009	451,077,252,328
pol_Latn	235,852,448,102	175,410,669	4,460,832,917	196,052,655,218	631,594,269,186
por_Latn	322,955,910,917	237,812,825	6,124,611,786	233,189,157,063	896,547,444,407
ron_Latn	92,755,690,867	65,876,383	1,696,970,479	76,264,228,246	250,658,132,448
slk_Latn	28,347,675,196	21,827,259	494,278,579	24,504,432,765	70,372,196,449
slv_Latn	13,480,602,614	10,277,173	238,644,943	11,867,536,246	35,258,183,993
spa_Latn	696,098,726,982	503,073,098	12,121,752,157	471,218,993,500	1,953,862,248,952
swe_Latn	93,095,597,524	66,812,562	1,754,677,064	75,784,600,156	251,109,959,822
cat_Latn	22,164,638,576	18,553,883	383,335,831	18,116,292,562	60,186,591,495
eus_Latn	2,187,606,572	1,974,218	37,621,611	2,034,478,450	6,052,165,410
glg_Latn	3,677,366,325	3,020,164	61,177,888	2,736,491,963	10,108,660,186
bos_Latn	18,404,991,480	14,613,088	268,156,648	14,828,824,339	46,070,953,520
isl_Latn	3,658,429,281	2,840,735	69,643,257	3,835,365,590	9,593,246,968
kat_Geor	5,248,761,917	3,335,164	63,722,098	4,538,769,891	10,155,612,392
mkd_Cyrl	4,353,674,682	3,565,647	57,008,331	3,406,651,991	9,439,624,767
als_Latn	6,144,592,594	5,385,262	95,101,980	5,892,424,412	16,095,653,237
srp_Cyrl	7,691,997,099	4,123,458	93,809,457	6,106,504,834	16,156,879,041
tur_Latn	105,086,134,521	84,541,414	1,941,885,324	85,625,744,754	283,639,575,889
ukr_Cyrl	83,197,910,551	47,395,787	1,169,038,372	60,690,550,123	182,867,693,190
nno_Latn nob_Latn	51,074,925,109	28,476,988	710,577,489	42,080,040,980	138,648,073,341
Total	**10,154,988,022,176**	**7,267,692,216**	**187,054,752,485**	**6,893,545,926,292**	**27,804,291,067,245**

Access Information

The primary download site for the data is hosted at the Norwegian national NIRD research data infrastructure, which offers premium connectivity to the Europen research data network. For convenience, selected subsets of the data have also been ingested to the Hugging Face Hub.

On select EuroHPC systems, the data is directly available for read-only access on the local filesytem:

- LUMI: `/appl/local/openeurollm/training/catalogue/hplt/2.0/`

Terms of Use

The HPLT terms of use distinguish between the *collection* and the *textual content*, where the first comprises the organization of the data and all metadata, and the latter the actual strings (the values of the JSON `text` fields) extracted from the original web documents. The collection is licensed under Creative Commons Public Domain (CC0) terms, whereas neither HPLT nor the original crawlers (IA and CC) hold rights to the textual content. HPLT has filtered out data that at the time of crawling likely was subject to standard opt-out procedures (the `robots.txt` protocol). A take-down mechanism is offered through the above download site. Users need to make sure that use of the data complies with any applicable legal framework, such as, among others, the EU Copyright Directive 2019/790 and the General Data Protection Regulation 2018, as amended.

Catalogue Curator

Stephan Oepen, University of Oslo, oe@ifi.uio.no

B. Analytics Report: Slovene in FineWeb 2.1.0

HPLT Analytics report

HPLTAnalytics

General overview

Corpus	Date	Language
fineweb-slv_Latn.parquet.tsv	7/28/2025	Slovenian (sl)

Volumes

Docs	Segments	Unique segments	Tokens	Characters	Size
12,059,130	272,812,432	166,819,250 (61.15 %)	7.8B	43,518,256,670	41.55 GB

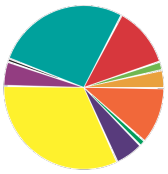
Top 10 domains

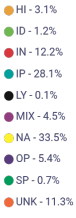
Domain	Docs	% of total
sta.si	287K	2.38%
delo.si	207K	1.72%
rtvslo.si	180K	1.49%
siol.net	178K	1.48%
dnevnik.si	161K	1.34%
metropolitan.si	142K	1.18%
zurnal24.si	132K	1.09%
blogspot.com	116K	0.97%
mojaobcina.si	84K	0.70%
mimovrste.com	83K	0.69%

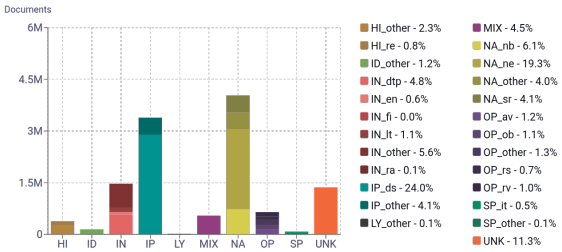
Top 10 TLDs

Domain	Docs	% of total
si	7.2M	60.02%
com	2.8M	22.82%
net	644K	5.34%
eu	384K	3.18%
org	373K	3.09%
info	138K	1.14%
cz	55K	0.46%
at	33K	0.28%
pt	27K	0.23%
tv	27K	0.23%

Register labels



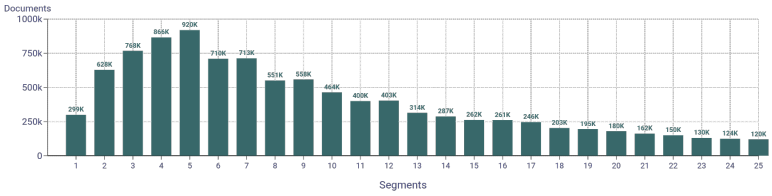




MT:7.8% | 935K Documents

Documents size (in segments)

<= 25 segments **82.18%** (9.9M documents)
> 25 segments **17.82%** (2.1M documents)

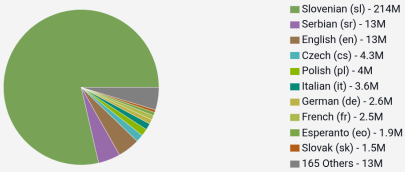


Documents by collection

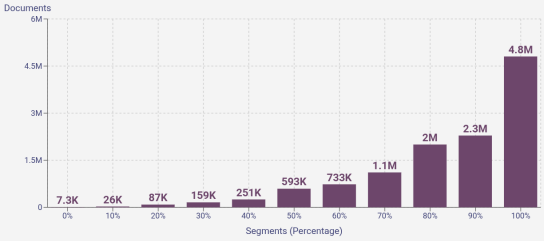
CC = 100.00%
IA = 0.00%

Language Distribution

Number of segments in the Slovenian (sl) corpus

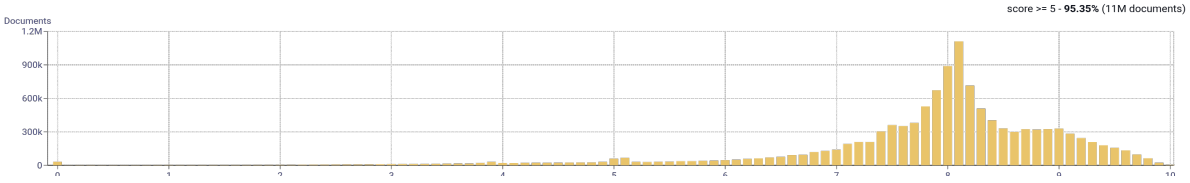


Percentage of segments in Slovenian (sl) inside documents



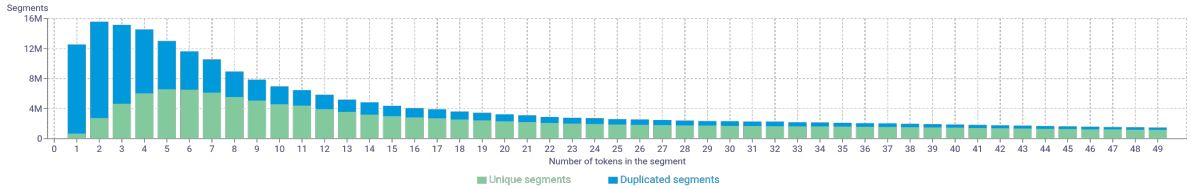
Distribution of documents by document score

score < 5 - **4.65%** (561K documents)
score >= 5 - **95.35%** (11M documents)

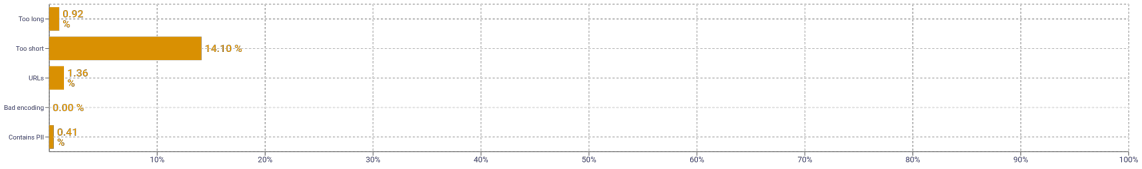


Segment length distribution by token

≤ **49** tokens = **225M** segments | **98M** duplicates
> **50** tokens = **48M** segments | **7.9M** duplicates



Segment noise distribution



Frequent n-grams

Size	n-grams
1	lahko | 37429784 · prostitutke | 10888736 · zelo | 8803224 · zato | 8036666 · veliko | 7507314
2	video posnetki | 2497622 · spletno mesto | 2146821 · porno video | 1320902 · spletna stran | 1130787 · erotična masaža | 1080920
3	mesto za zmenke | 1289004 · stran za zmenke | 777964 · porno video posnetki | 751666 · mesta za zmenke | 611084 · brezplačno spletno mesto | 507057
4	spletno mesto za zmenke | 1274554 · spletna mesta za zmenke | 518295 · spletna stran za zmenke | 405060 · spletno stran za zmenke | 164210 · spletišč in mobilnih aplikacij | 132386
5	brezplačno spletno mesto za zmenke | 365782 · brezplačna spletna stran za zmenke | 172385 · brezplačna spletna mesta za zmenke | 155264 · dostopnosti spletišč in mobilnih aplikacij | 132200 · novica je dostopna le naročnikom | 129968

About HPLT Analytics

Volumes - Segments
Segments correspond to paragraph and list boundaries as defined by HTML elements (<p>, <ul>, <ol>, etc.) replaced by newlines.

Volumes - Tokens
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md

Type-Token Ratio
Lexical variety computed as *number or types (uniques)/number of tokens*, after removing punctuation (https://www.sltinfo.com/wp-content/uploads/2014/01/type-token-ratio.pdf).

Document size (in segments)
Segments correspond to paragraph and list boundaries as defined by HTML elements (<p>, <ul>, <ol>, etc.) replaced by newlines.

Language distribution
Language identified with FastSpell (https://github.com/mbanon/fastspell).

Distribution of segments by fluency score
Obtained with Monocleaner (https://github.com/bitextor/monocleaner).

Distribution of documents by average fluency score
Obtained with Monocleaner (https://github.com/bitextor/monocleaner).

Distribution of documents by document score
Obtained with Web Docs Scorer (https://github.com/pablop16n/web-docs-scorer/).

Segment length distribution by token
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md

Segment noise distribution
Obtained with Bicleaner Hardrules (https://github.com/bitextor/bicleaner-hardrules/).

Frequent n-grams
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md, after removing n-grams starting or ending in a stopword. Stopwords from https://github.com/hplt-project/data-analytics-tool/blob/main/scripts/resources/README.txt

Register labels

Name	Abbr.
Machine-translated	MT
Lyrical	LY
Spoken	SP
Interview	it
Interactive discussion	ID
Narrative	NA
News report	ne
Sports report	sr
Narrative blog	nb

Name	Abbr.
How-to or instructions	HI
Recipe	re
Informational persuasion	IP
Description with intent to sell	ds
News & opinion blog or editorial	ed
Informational description	IN
Enciclopedia article	en
Research article	ra

Name	Abbr.
Description of a thing or person	dtp
FAQ	fi
Legal terms & conditions	lt
Opinion	OP
Review	rv
Opinion blog	ob
Denominational religious blog or sermon	rs
Advice	av

C. Analytics Report: Slovene in HPLT 2.0

HPLT Analytics report

HPLTAnalytics

General overview

Corpus	Date	Language
slv_Latn.jsonl.tsv	6/7/2025	Slovenian (sl)

Volumes

Docs	Segments	Unique segments	Tokens	Characters	Size
10,277,172	238,597,185	107,466,653 (45.04 %)	6.4B	35,029,805,659	33.47 GB

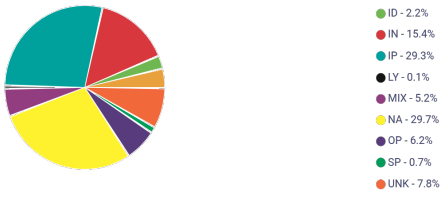
Top 10 domains

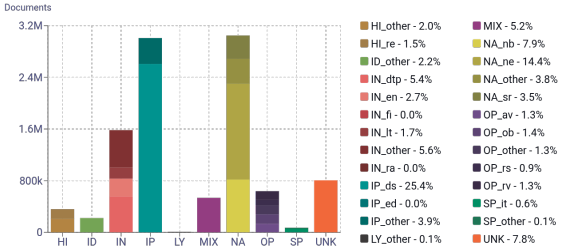
Domain	Docs	% of total
wikipedia.org	276K	2.69%
sta.si	226K	2.20%
siol.net	141K	1.37%
delo.si	130K	1.26%
blogspot.com	120K	1.16%
dnevnik.si	83K	0.81%
metropolitan.si	77K	0.75%
slo-tech.com	73K	0.71%
rtvslo.si	72K	0.70%
agoda.com	71K	0.69%

Top 10 TLDs

Domain	Docs	% of total
si	6.3M	61.21%
com	2.2M	21.01%
net	558K	5.43%
org	555K	5.40%
eu	252K	2.45%
info	120K	1.17%
se	28K	0.27%
je	24K	0.24%
tv	22K	0.21%
at	19K	0.19%

Register labels

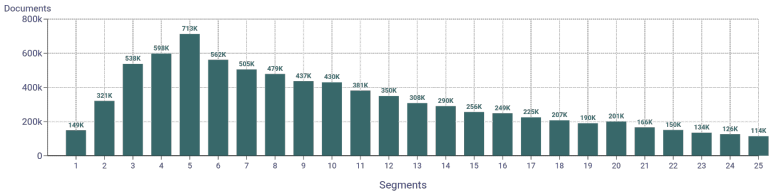




MT:5.0% | 514K Documents

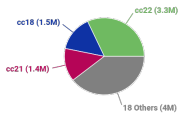
Documents size (in segments)

<= 25 segments **78.6%** (8.1M documents)
> 25 segments **21.4%** (2.2M documents)



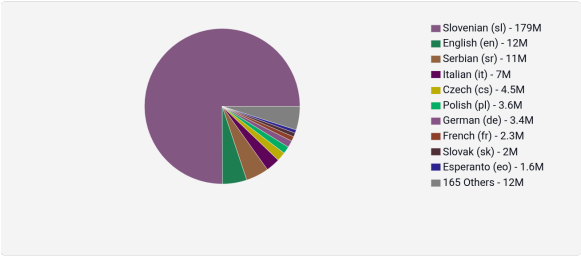
Documents by collection

CC = 70.36%
IA = 29.64%

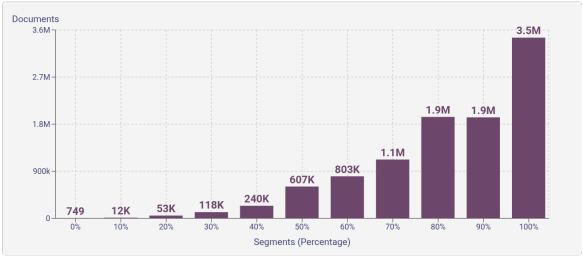


Language Distribution

Number of segments in the Slovenian (sl) corpus

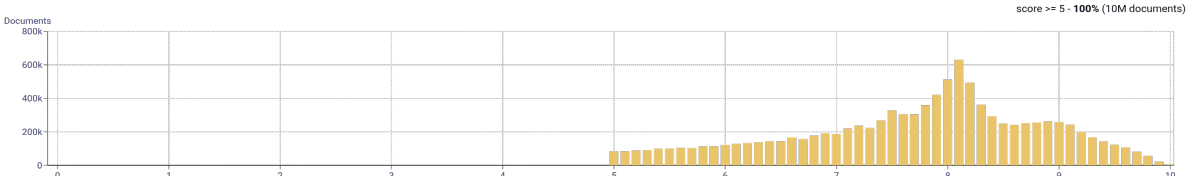


Percentage of segments in Slovenian (sl) inside documents



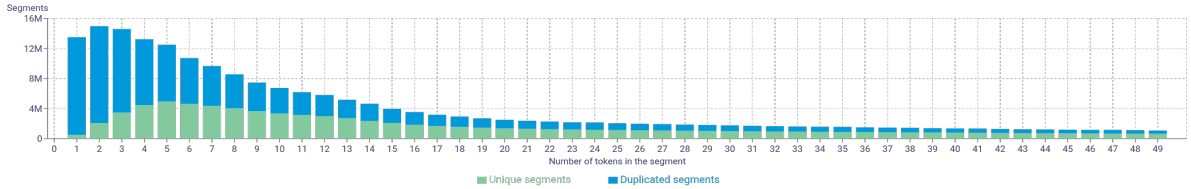
Distribution of documents by document score

score < 5 - **0%** (0 documents)
score >= 5 - **100%** (10M documents)

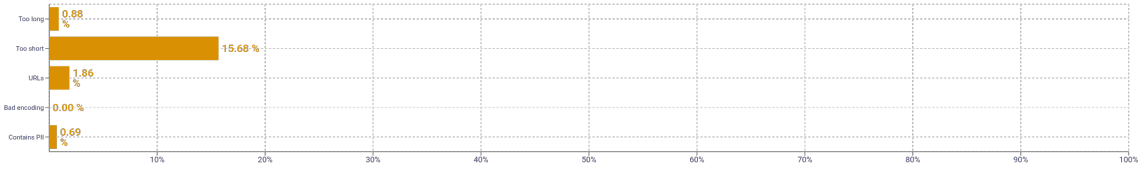


Segment length distribution by token

≤ **49** tokens = **200M** segments | **116M** duplicates
> **50** tokens = **39M** segments | **16M** duplicates



Segment noise distribution



Frequent n-grams

Size	n-grams
1	lahko | 25982227 · zelo | 6691629 · a | 6631583 · leta | 6410881 · zato | 6406868
2	uredi kodo | 974504 · spletni strani | 834745 · republike slovenije | 684698 · spletne strani | 655489 · osebnih podatkov | 540481
3	uradni list rs | 388741 · člena tega zakona | 187426 · državna revizijska komisija | 166992 · dodaj v košarico | 159247 · odstavka tega člena | 157674
4	celotna novica je dostopna | 136900 · evropskega parlamenta in sveta | 64408 · e-poštni naslov je zaščiten | 55487 · obvezno pokojninsko in invalidsko | 48545 · uradnem listu republike slovenije | 47721
5	novica je dostopna le naročnikom | 136880 · naslov je zaščiten proti smetenju | 55436 · kazensko odgovoren za javno spodbujanje | 44940 · posameznik kazensko odgovoren za javno | 44926 · odgovoren za javno spodbujanje sovraštva | 44788

About HPLT Analytics

Volumes - Segments
Segments correspond to paragraph and list boundaries as defined by HTML elements (<p>, <ul>, <ol>, etc.) replaced by newlines.

Volumes - Tokens
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md

Type-Token Ratio
Lexical variety computed as *number or types (uniques)/number of tokens*, after removing punctuation (https://www.sltinfo.com/wp-content/uploads/2014/01/type-token-ratio.pdf).

Document size (in segments)
Segments correspond to paragraph and list boundaries as defined by HTML elements (<p>, <ul>, <ol>, etc.) replaced by newlines.

Language distribution
Language identified with FastSpell (https://github.com/mbanon/fastspell).

Distribution of segments by fluency score
Obtained with Monocleaner (https://github.com/bitextor/monocleaner).

Distribution of documents by average fluency score
Obtained with Monocleaner (https://github.com/bitextor/monocleaner).

Distribution of documents by document score
Obtained with Web Docs Scorer (https://github.com/pablop16n/web-docs-scorer/).

Segment length distribution by token
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md

Segment noise distribution
Obtained with Bicleaner Hardrules (https://github.com/bitextor/bicleaner-hardrules/).

Frequent n-grams
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md, after removing n-grams starting or ending in a stopword. Stopwords from https://github.com/hplt-project/data-analytics-tool/blob/main/scripts/resources/README.txt

Register labels

Name	Abbr.
Machine-translated	MT
Lyrical	LY
Spoken	SP
Interview	it
Interactive discussion	ID
Narrative	NA
News report	ne
Sports report	sr
Narrative blog	nb

Name	Abbr.
How-to or instructions	HI
Recipe	re
Informational persuasion	IP
Description with intent to sell	ds
News & opinion blog or editorial	ed
Informational description	IN
Enciclopedia article	en
Research article	ra

Name	Abbr.
Description of a thing or person	dtp
FAQ	fi
Legal terms & conditions	lt
Opinion	OP
Review	rv
Opinion blog	ob
Denominational religious blog or sermon	rs
Advice	av

D. Analytics Report: Irish in FineWeb 2.1.0

HPLT Analytics report

HPLTAnalytics

General overview

Corpus	Date	Language
fineweb-gle_Latn.parquet.tsv	7/28/2025	Irish (ga)

Volumes

Docs	Segments	Unique segments	Tokens	Characters	Size
646,842	13,307,836	9,777,245 (73.47 %)	417M	2,182,223,496	2.16 GB

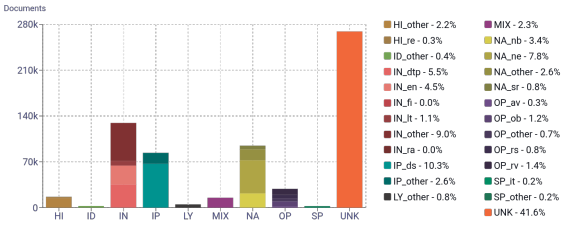
Top 10 domains

Domain	Docs	% of total
tuairisc.ie	26K	4.00%
wikipedia.org	24K	3.64%
europa.eu	16K	2.50%
airbnb.ie	15K	2.35%
focloir.ie	12K	1.82%
duchas.ie	11K	1.72%
rte.ie	8.7K	1.34%
martech.zone	7.4K	1.14%
blogspot.com	7.3K	1.13%
eferrit.com	6.5K	1.01%

Top 10 TLDs

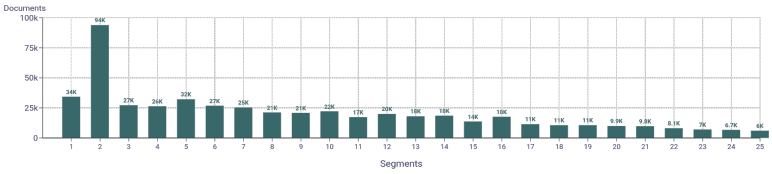
Domain	Docs	% of total
com	230K	35.61%
ie	193K	29.87%
org	47K	7.30%
eu	21K	3.20%
net	12K	1.90%
pt	9.5K	1.47%
xyz	7.8K	1.20%
monster	7.4K	1.14%
zone	7.4K	1.14%
gov.ie	7.1K	1.09%

Register labels



MT:39.4% | 255K Documents

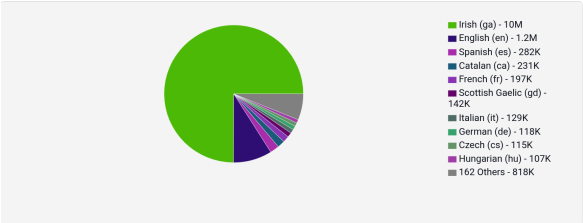
Documents size (in segments)



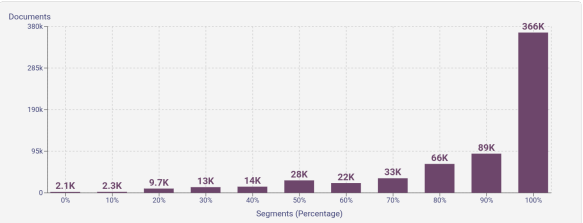
Documents by collection

Language Distribution

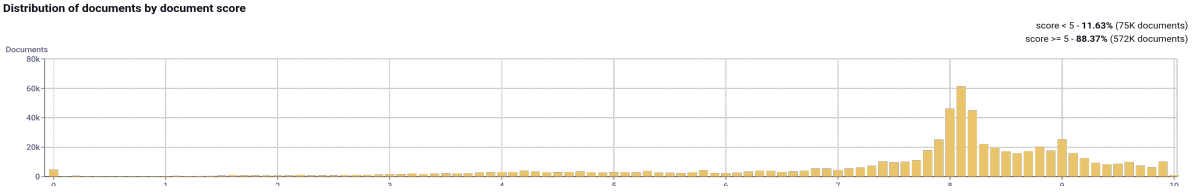
Number of segments in the Irish (ga) corpus



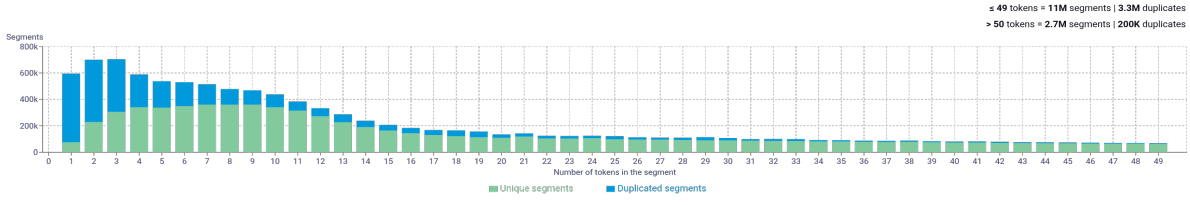
Percentage of segments in Irish (ga) inside documents



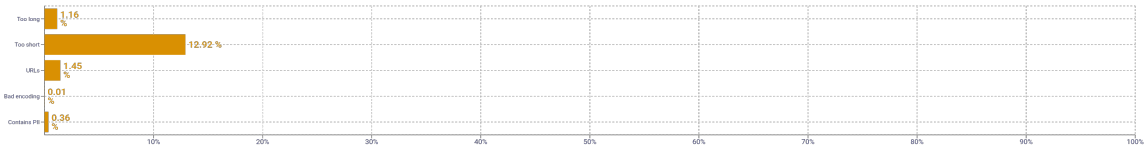
Distribution of documents by document score



Segment length distribution by token



Segment noise distribution



Frequent n-grams

Size	n-grams
1	bhfuil | 1947971 · atá | 1541384 · d | 1468033 · féidir | 1458046 · níos | 1261204
2	féidir leat | 477039 · níos mó | 454548 · of the | 117005 · níos fearr | 114334 · bhaint amach | 92961
3	saor in aisce | 274784 · chuid is mó | 122667 · nuair a bhíonn | 76610 · chur ar fáil | 62748 · fud an domhain | 61480
4	lá atá inniu ann | 30144 · rud é go bhfuil | 19951 · níos forbartha le fáil | 16205 · fáil i dteanga eile | 16205 · cuidiú leis an vicipéid | 16165
5	forbartha le fáil i dteanga | 16205 · alt níos forbartha le fáil | 16205 · leat aistriúchán gaeilge a dhéanamh | 15891 · ós rud é go bhfuil | 15545 · rud a fhágann go bhfuil | 13627

About HPLT Analytics

Volumes - Segments
Segments correspond to paragraph and list boundaries as defined by HTML elements (<p>, <ul>, <ol>, etc.) replaced by newlines.

Volumes - Tokens
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md

Type-Token Ratio
Lexical variety computed as *number or types (uniques)/number of tokens*, after removing punctuation (https://www.sltinfo.com/wp-content/uploads/2014/01/type-token-ratio.pdf).

Document size (in segments)
Segments correspond to paragraph and list boundaries as defined by HTML elements (<p>, <ul>, <ol>, etc.) replaced by newlines.

Language distribution
Language identified with FastSpell (https://github.com/mbanon/fastspell).

Distribution of segments by fluency score
Obtained with Monocleaner (https://github.com/bitextor/monocleaner).

Distribution of documents by average fluency score
Obtained with Monocleaner (https://github.com/bitextor/monocleaner).

Distribution of documents by document score
Obtained with Web Docs Scorer (https://github.com/pablop16n/web-docs-scorer/).

Segment length distribution by token
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md

Segment noise distribution
Obtained with Bicleaner Hardrules (https://github.com/bitextor/bicleaner-hardrules/).

Frequent n-grams
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md, after removing n-grams starting or ending in a stopword. Stopwords from https://github.com/hplt-project/data-analytics-tool/blob/main/scripts/resources/README.txt

Register labels

Name	Abbr.
Machine-translated	MT
Lyrical	LY
Spoken	SP
Interview	it
Interactive discussion	ID
Narrative	NA
News report	ne
Sports report	sr
Narrative blog	nb

Name	Abbr.
How-to or instructions	HI
Recipe	re
Informational persuasion	IP
Description with intent to sell	ds
News & opinion blog or editorial	ed
Informational description	IN
Enciclopedia article	en
Research article	ra

Name	Abbr.
Description of a thing or person	dtp
FAQ	fi
Legal terms & conditions	lt
Opinion	OP
Review	rv
Opinion blog	ob
Denominational religious blog or sermon	rs
Advice	av

E. Analytics Report: Irish in HPLT 2.0

HPLT Analytics report

HPLTAnalytics

General overview

Corpus	Date	Language
gle_Latn.jsonl.tsv	9/16/2024	Irish (ga)

Volumes

Docs	Segments	Unique segments	Tokens	Characters	Size
490,787	10,993,158	5,866,989 (53.37 %)	336M	1,738,847,965	1.72 GB

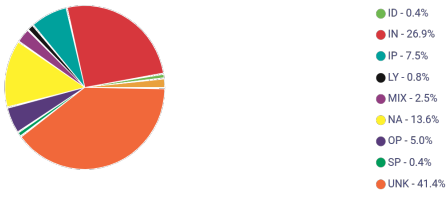
Top 10 domains

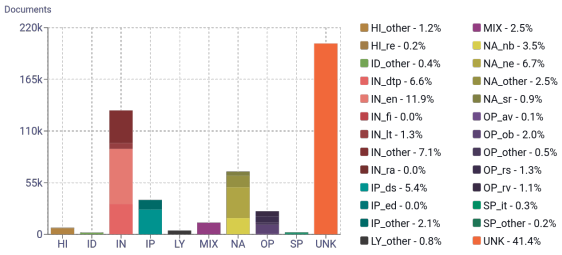
Domain	Docs	% of total
wikipedia.org	63K	12.94%
tuairisc.ie	25K	5.11%
europa.eu	11K	2.28%
stealthsettings...	8.4K	1.71%
blogspot.com	8.3K	1.69%
soft-free-downl...	8.1K	1.65%
duchas.ie	7.8K	1.60%
itsmygame.org	6.9K	1.40%
daily-helper.com	6.6K	1.35%
nos.ie	6.5K	1.32%

Top 10 TLDs

Domain	Docs	% of total
com	179K	36.49%
ie	149K	30.43%
org	88K	18.00%
eu	15K	2.98%
net	14K	2.78%
mobi	5.2K	1.07%
pt	3.6K	0.74%
gov.ie	2.9K	0.60%
at	1.8K	0.36%
ru	1.7K	0.35%

Register labels

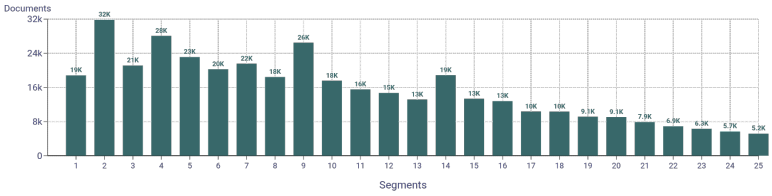




MT:38.9% | 191K Documents

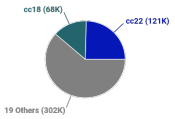
Documents size (in segments)

<= 25 segments **78.74%** (386K documents)
> 25 segments **21.26%** (104K documents)



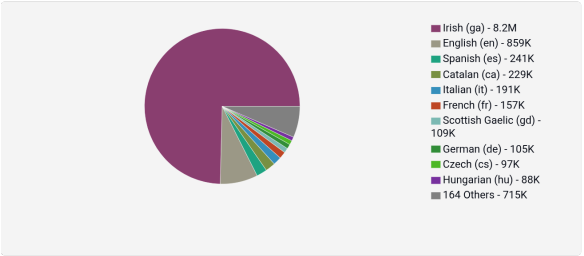
Documents by collection

CC = 59.98%
IA = 40.02%

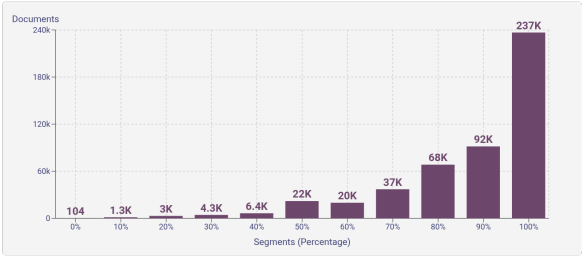


Language Distribution

Number of segments in the Irish (ga) corpus

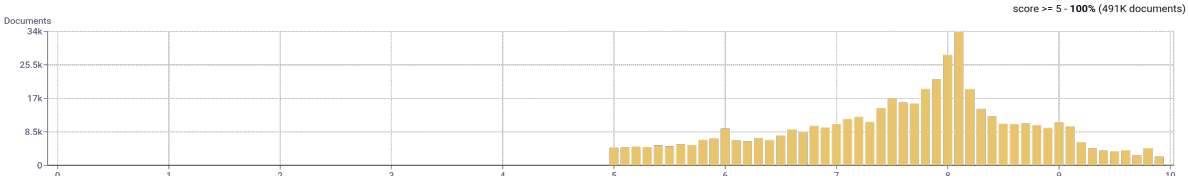


Percentage of segments in Irish (ga) inside documents



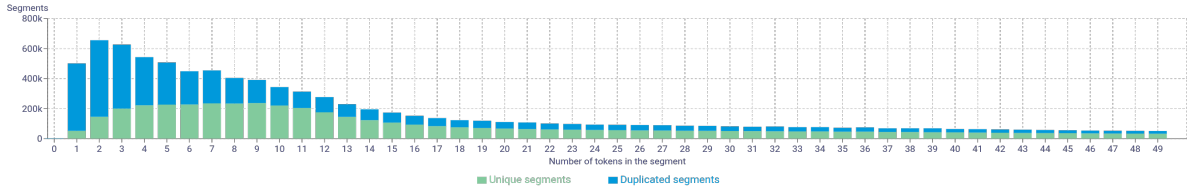
Distribution of documents by document score

score < 5 - **0%** (0 documents)
score >= 5 - **100%** (491K documents)

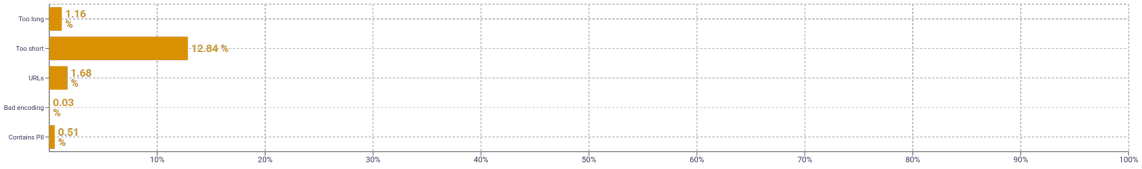


Segment length distribution by token

≤ **49** tokens = **8.9M** segments | **4.4M** duplicates
> **50** tokens = **2.1M** segments | **754K** duplicates



Segment noise distribution



Frequent n-grams

Size	n-grams
1	sin | 1565921 · bhfuil | 1395697 · bhí | 1241338 · atá | 1195774 · d | 1123479
2	nios mó | 301320 · féidir leat | 290291 · átha cliath | 91311 · nios fearr | 71883 · sin féin | 70114
3	saor in aisce | 208120 · chuid is mó | 89530 · chur ar fáil | 50016 · fud an domhain | 47986 · nuair a bhí | 45820
4	lá atá inniu ann | 23173 · line saor in aisce | 19579 · rud é go bhfuil | 13083 · roinnt i lionraí sóisialta | 12285 · rud é nach bhfuil | 11809
5	dearmad a mheas an cluiche | 10997 · play ar line a flash | 10954 · ós rud é go bhfuil | 9689 · más rud é nach bhfuil | 8855 · más mian leat an cluiche | 5850

About HPLT Analytics

Volumes - Segments
Segments correspond to paragraph and list boundaries as defined by HTML elements (<p>, <ul>, <ol>, etc.) replaced by newlines.

Volumes - Tokens
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md

Type-Token Ratio
Lexical variety computed as *number or types (uniques)/number of tokens*, after removing punctuation (https://www.sltinfo.com/wp-content/uploads/2014/01/type-token-ratio.pdf).

Document size (in segments)
Segments correspond to paragraph and list boundaries as defined by HTML elements (<p>, <ul>, <ol>, etc.) replaced by newlines.

Language distribution
Language identified with FastSpell (https://github.com/mbanon/fastspell).

Distribution of segments by fluency score
Obtained with Monocleaner (https://github.com/bitextor/monocleaner).

Distribution of documents by average fluency score
Obtained with Monocleaner (https://github.com/bitextor/monocleaner).

Distribution of documents by document score
Obtained with Web Docs Scorer (https://github.com/pablop16n/web-docs-scorer/).

Segment length distribution by token
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md

Segment noise distribution
Obtained with Bicleaner Hardrules (https://github.com/bitextor/bicleaner-hardrules/).

Frequent n-grams
Tokenized with https://github.com/hplt-project/data-analytics-tool/blob/main/tokenizers-info.md, after removing n-grams starting or ending in a stopword. Stopwords from https://github.com/hplt-project/data-analytics-tool/blob/main/scripts/resources/README.txt

Register labels

Name	Abbr.
Machine-translated	MT
Lyrical	LY
Spoken	SP
Interview	it
Interactive discussion	ID
Narrative	NA
News report	ne
Sports report	sr
Narrative blog	nb

Name	Abbr.
How-to or instructions	HI
Recipe	re
Informational persuasion	IP
Description with intent to sell	ds
News & opinion blog or editorial	ed
Informational description	IN
Enciclopedia article	en
Research article	ra

Name	Abbr.
Description of a thing or person	dtp
FAQ	fi
Legal terms & conditions	lt
Opinion	OP
Review	rv
Opinion blog	ob
Denominational religious blog or sermon	rs
Advice	av

F. Example Download Script: FineWeb 1.4.0

fineweb/1.4.0/etc/download.slurm

```
#!/bin/bash

#SBATCH --job-name=download
#SBATCH --partition=small
#SBATCH --account=project_462000131
#SBATCH --time=72:00:00
#SBATCH --mail-type=FAIL
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=8
#SBATCH --mem-per-cpu=2G

#
# determine our root directory and load a common preabmle
#
if [ -n "${SLURM_JOB_ID}" ]; then
  script=$(scontrol show job ${SLURM_JOB_ID} \
  | awk '/^ *Command=/ {sub(/^ *Command=/, "", $1); print $1;}');
  path=$(dirname ${script});
  ROOT="${path%/etc}";
else
  path=$(dirname $0);
  if [ "${path#./}" != "${path}" ]; then
    path="$(pwd)/${path#./}";
  fi
  if [ "${path#/}" = "${path}" ]; then
    if [ "${path}" = "." ]; then
      path="$(pwd)";
    else
      path="$(pwd)/${path}";
    fi
  fi
  ROOT="${path%/etc}";
fi
source ${ROOT}/../../etc/preamble.sh;

module reset;
module load LUMI cURL cray-python;

DATA=edu;
TARGET="HuggingFaceFW/fineweb-edu";
REVISION="v1.4.0";
HF="https://huggingface.co/datasets/${TARGET}/resolve/${REVISION}/";
```

```
cd ${ROOT};

echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] download.slurm: root in ${ROOT}; target: ${DATA}."
echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] download.slurm: using ${CORES} cores."

if [ ! -d ./${DATA} ]; then mkdir ./${DATA}; fi
cd ./${DATA}

CURL="curl --no-progress-meter --show-error --create-dirs --location";
CURL="${CURL} --globoff --remote-time";
CURL="${CURL} --fail --retry 5 --remove-on-error";
CORES=${SLURM_CPUS_ON_NODE:-8};
CURL="${CURL} --parallel --parallel-max ${CORES}";
CURL="${CURL} --rate 2/s";

URL=all.url;
LOG=curl.log;
cat /dev/null > ${URL};
cat /dev/null > ${LOG};

#
# determine list of data files and create download map
#
python -c "
import sys;
sys.path.append(\"${BASE}/etc\");
import huggingface as hf;
hf.api_retrieve_files(\"${TARGET}\", \"${REVISION}\", \"^data/\");
" \
| while read line; do \
  target=${line#data/};
  if [ ! -f ${target} ]; then
    echo "url ${HF}${line}" >> ${URL};
    echo "output ${target}" >> ${URL};
  fi;
done;

echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] download.slurm: URL directives (${URL}):"
cat ${URL};
echo;
exit;
#
# enable detailed download statistics
#
CURL="${CURL} --write-out \"%{stderr}{\\\"content-type\\\": \\\"%header{content-type}\\\", \\\"conten
```

```
#
# now download all files that are missing or appear out of date
#
eval ${CURL} --config ${URL} 2>> ${LOG};

#
# make sure everything is world-readable (and directories searchable)
#
chmod -R o+rX .;

echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] download.slurm: all downloads complete."
exit 0;
```

G. Example Validation Script: HPLT 2.0

hplt/2.0/etc/validate.slurm

```
#!/bin/bash

#SBATCH --job-name=validate
#SBATCH --partition=small
#SBATCH --account=project_462000131
#SBATCH --time=72:00:00
#SBATCH --mail-type=FAIL
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=128
#SBATCH --mem-per-cpu=1G

#
# determine our root directory and load a common preabmle
#
if [ -n "${SLURM_JOB_ID}" ]; then
  script=$(scontrol show job ${SLURM_JOB_ID} \
  | awk '/^ *Command=/ {sub(/^ *Command=/, "", $1); print $1;}');
  path=$(dirname ${script});
  ROOT="${path%/etc}";
else
  path=$(dirname $0);
  if [ "${path#./}" != "${path}" ]; then
    path="$(pwd)/${path#./}";
  fi
  if [ "${path#/}" = "${path}" ]; then
    if [ "${path}" = "." ]; then
      path="$(pwd)";
    else
      path="$(pwd)/${path}";
    fi
  fi
  ROOT="${path%/etc}";
fi
source ${ROOT}/../../etc/preamble.sh;

module reset;
module load LUMI cURL parallel;

DATA=cleaned;

cd ${ROOT};
```

```
echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] validate.slurm: root in ${ROOT}; target: ${DATA}."
echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] validate.slurm: using ${CORES} cores."

if [ ! -d ./${DATA} ]; then mkdir ./${DATA}; fi
cd ./${DATA}

CURL="curl --disable --no-progress-meter --show-error --create-dirs --location";
CURL="${CURL} --fail --retry 2";
CORES=${SLURM_CPUS_ON_NODE:-8};
CURL="${CURL} --parallel --parallel-max ${CORES}";

MD5=all.md5;
LOG=md5sum.log;
cat /dev/null > ${MD5};
cat /dev/null > ${LOG};

#
# create fresh download maps for the checksum files
#
${CURL} https://data.hplt-project.org/two/${DATA}/hplt_monolingual_map_${DATA}_2.0.txt \
| while read line; do \
  prefix=${line%/*/*};
  target=${line#${prefix}/};
  path=${target%/*};
  file=${target#${path}/};
  #
  # ignore existing files; we always want to use up-to-date checksums
  #
  /bin/rm -f ${path}/.${file}.md5;
  echo "url ${line}.md5" >> ${MD5};
  echo "output ${path}/.${file}.md5" >> ${MD5};
done;

echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] validate.slurm: MD5 directives (${MD5}):"
cat ${MD5};
echo;

#
# now make sure we have current checksum files for everything
#
eval ${CURL} --config ${MD5}

function validate() {
  file=${1};
  path=${file%/*};
  file=${file#${path}/};
```

```
  if [ ! -d ${path} ]; then
    echo "invalid sub-directory: ${path}." >&2;
    return 1;
  fi
  (
    cd ${path};
    target=$(awk '{print $2}' ${file});
    if [ ! -f ${target} ]; then
      echo "missing target file: ${path}/${target#./}." >&2;
      return 1;
    fi
    if md5sum --check --status ${file}; then
      echo "check-sum validated: ${path}/${target#./}." >&2;
    else
      echo "check-sum failure; removing: ${path}/${target#./}." >&2;
      /bin/rm ${target};
    fi
  )
} # validate()
export -f validate;

PARALLEL="parallel --plain --will-cite";
PARALLEL="${PARALLEL} --group --jobs ${CORES}";

#
# finally, iterate through all the check-sum files
#
egrep "^output " ${MD5} | sed "s/output //" \
| eval ${PARALLEL} validate {};

#
# make sure everything is world-readable (and directories searchable)
#
chmod -R o+rX .;

echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] validate.slurm: all files processed."
```

H. Example Statistics Script: MADLAD-400 1.0

madlad-400/1.0/etc/index.slurm

```
#!/bin/bash

#SBATCH --job-name=index
#SBATCH --partition=small
#SBATCH --account=project_462000131
#SBATCH --time=72:00:00
#SBATCH --mail-type=FAIL
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=16
#SBATCH --mem-per-cpu=16380M

#
# determine our root directory and load a common preabmle
#
if [ -n "${SLURM_JOB_ID}" ]; then
  script=$(scontrol show job ${SLURM_JOB_ID} \
  | awk '/^ *Command=/ {sub(/^ *Command=/, "", $1); print $1;}');
  path=$(dirname ${script});
  ROOT="${path%/etc}";
else
  path=$(dirname $0);
  if [ "${path#./}" != "${path}" ]; then
    path="$(pwd)/${path#./}";
  fi
  if [ "${path#/}" = "${path}" ]; then
    if [ "${path}" = "." ]; then
      path="$(pwd)";
    else
      path="$(pwd)/${path}";
    fi
  fi
  ROOT="${path%/etc}";
fi
source ${ROOT}/../../etc/preamble.sh;

module reset;
module load cray-python;
source ${BASE}/python/bin/activate;

DATA=clean;

cd ${ROOT};
```

```
echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] index.slurm: root in ${ROOT}; target: ${DATA}."
echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] index.slurm: using ${CORES} cores."

#
# command-line arguments can be directories; otherwise glob ourselves
#
for i in ${*:-*}; do
  path=$(readlink -e ${i});
  if [ -z "${path}" ]; then continue; fi
  /usr/bin/time -v python -c "
import sys;
sys.path.append(\"${BASE}/etc\");
import index;
index.index_directory(\"${path}\", pattern = \".jsonl.gz\", cores = ${CORES}, url = None);
  "
done;

#
# make sure everything is world-readable (and directories searchable)
#
chmod -R o+rX .;

echo "[$(date +"%Y-%m-%d (%H:%M:%S)")] count.slurm: all files processed."
exit 0;
```

etc/index.py

```
import io;
import glob;
import gzip;
import hashlib;
import json;
import multiprocessing as mp;
from operator import itemgetter;
import os;
import re;
import sys;
import time;
from urllib.parse import urlsplit;
import zstandard as zstd;

def index_file(path, text = "text", url = "u", level = 1):

  stream = None;
  if path.endswith(".zst") or path.endswith(".zstd"):
```

```
  decompressor = zstd.ZstdDecompressor();
  stream = decompressor.stream_reader(open(path, "rb"));
  stream = io.TextIOWrapper(stream, encoding = "utf-8", errors = "replace");
elif path.endswith(".gz"):
  stream = gzip.open(path, mode = "rt", encoding = "utf-8", errors = "replace");
else:
  print(f"index_file(): invalid input format {path}; exit.",
        file = sys.stderr)
  exit(1);

signatures = dict();
if url is not None: domains = dict(); urls = dict();
else: domains = urls = None;
normalize = re.compile(r"\W", re.IGNORECASE);
directory, file = os.path.split(path);
key = os.path.join(os.path.sep.join(directory.split(os.path.sep)[-level:]), file);

def index(item, dictionary):
  if item in dictionary:
    _ = dictionary[item];
    _["n"] += 1;
    _[key].append(i);
  else:
    dictionary[item] = {"n": 1, key: [i]};

for i, line in enumerate(stream):
  try:
    _ = json.loads(line);
    document = normalize.sub("", _[text]).lower();
    signature = hashlib.md5(document.encode("utf-8")).hexdigest();
    if url is not None:
      address = _[url];
      domain = urlsplit(address).netloc;
  except Exception as error:
    print(f"index_file(): #{i} decoding error: {error}.",
          file = sys.stderr);
    continue;
  index(signature, signatures);
  if url is not None:
    index(domain, domains);
    index(address, urls);

def output(dictionary, suffix):
  name = os.path.join(directory, "." + file + suffix + ".zst");
  compressor = zstd.ZstdCompressor(level = 10, threads = 1);
  stream = compressor.stream_writer(open(name, "wb"));
```

```
    stream = io.TextIOWrapper(stream, encoding = "utf-8", errors = "replace");
    for key, value in sorted(dictionary.items()):
      print("{}\t{}".format(key, value["n"]),
            end = "\t", file = stream);
      value.pop("n");
      json.dump(value, stream);
      print(file = stream);
    stream.close();

  for _ in (".zstd", ".zst", ".gz", ".jsonl", ".json"):
    if file.endswith(_): file = file[:-len(_)];
  output(signatures, ".signatures");
  if url is not None:
    output(domains, ".domains");
    output(urls, ".urls");

  return i + 1;

def index_directory(path, pattern = r"\.jsonl\.zst$", cores = 1,
                    text = "text", url = "u", level = 1, tree = False):

  def walk(path, pattern, tree):
    block = re.compile(r"/\.(?:domains|urls|signatures)\.zst$");
    if not os.path.isdir(path):
      print(f"merge.py: ignoring invalid path {path}.",
            file = sys.stderr);
      return [];
    result = [];
    for path in glob.glob(os.path.join(path, "*"), include_hidden = True):
      if tree and os.path.isdir(path):
        result += walk(path, pattern, tree);
      elif os.path.isfile(path) and pattern.search(path):
        if not block.search(path): result.append(path);
    return result;

  start = time.time();
  pattern = re.compile(pattern);
  files = walk(path, pattern, tree);
  print("index.py: reading {}.".format([file[len(path) + 1:] for file in files]));
  with mp.Pool(cores) as pool:
    counts = pool.starmap(index_file,
                          ((file, text, url, level) for file in files));
  print("index.py: processed {} files; {} documents; {:.2f} seconds."
        "".format(len(counts), sum(counts), time.time() - start));

  def compress(suffix):
```

```
    #
    # create a compressed output stream
    #
    name = os.path.join(path, "." + suffix + ".zst");
    compressor = zstd.ZstdCompressor(level = 10, threads = cores);
    stream = compressor.stream_writer(open(name, "wb"));
    stream = io.TextIOWrapper(stream, encoding = "utf-8", errors = "replace");
    return stream;

  n = r = 0;
  for key in ["domains", "urls", "signatures"] if url is not None else ["signatures"]:
    pattern = re.compile(r"\.[^/]+\." + key + ".zst$");
    files = walk(path, pattern, tree);
    print("index.py: merging {}.".format([file[len(path) + 1:] for file in files]));
    inputs = connect(files);
    n += len(inputs);
    output = compress(key);
    r += merge(inputs, output);
    output.close();
    for _ in inputs: _["stream"].close();
  print("index.py: merged {} files; {} records; {:.2f} seconds."
        "".format(n, r, time.time() - start));

def connect(files):
  #
  # open the individual index files and read their first entry
  #
  inputs = [];
  for file in files:
    decompressor = zstd.ZstdDecompressor();
    stream = decompressor.stream_reader(open(file, "rb"));
    stream = io.TextIOWrapper(stream, encoding = "utf-8", errors = "replace");
    input = {"stream": stream, "file": file, "n": 0};
    key, count, input = parse(input);
    if key is not None: inputs.append((key, count, input));
  return inputs;

def parse(input):
  #
  # parse one tab-separated entry from an index file
  #
  line = next(input["stream"], None);
  if line is None:
    input["stream"].close();
    return None, None, input;
  input["n"] += 1;
```

```
  try:
    _ = line.find("\t");
    key = line[:_];
    line = line[_ + 1:];
    _ = line.find("\t");
    count = int(line[:_]);
    entry = json.loads(line[_ + 1:]);
  except Exception as error:
    print("index.py: aborting input from {}, #{}: {}."
          "".format(input["file"], input["n"], error),
          file = sys.stderr);
    input["stream"].close();
    return None, None, input;
  input["entry"] = entry;
  return key, count, input;

def match(queue, key, counts):
  count = 0;
  while len(queue) and queue[0][0] == key:
    match = queue.pop(0);
    count += match[1];
    match = parse(match[2]);
    if match[0] is None:
      counts["n"] += match[2]["n"];
      continue;
    else: queue.append(match);
  return count;

def merge(inputs, stream):
  #
  # sorted merge of records from a set of input streams
  #
  n = 0;
  while len(inputs):
    #
    # _fix_me_ should use a genuine priority queue
    #
    inputs.sort(key = itemgetter(0));
    key, count, input = inputs.pop(0);

    #
    # process other (currently visible) entries with the same key
    #
    while len(inputs) and inputs[0][0] == key:
      #
      # merge count and payload of matching entry
```

```
      #
      match = inputs.pop(0);
      count += match[1];
      input["entry"].update(match[2]["entry"]);
      #
      # update for next record and re-queue, unless exhausted
      #
      match = parse(match[2]);
      if match[0] is None: continue;
      else: inputs.append(match);
    print(f"{key}\t{count}", end = "\t", file = stream);
    json.dump(input["entry"], stream);
    print(file = stream);
    n += 1;
    #
    # get next key, count, and entry from this input file;
    # re-insert into the priority queue, unless exhausted
    #
    key, count, input = parse(input);
    if key is None: continue;
    else: inputs.append((key, count, input));

  return n;

def intersect(left, right, verbose = False, level = 2):

  def drain(queue, counts):
    n = 0;
    while len(queue):
      key, count, entry = queue.pop(0);
      while key is not None:
        n += count;
        key, count, entry = parse(entry);
      counts["n"] += entry["n"];
    return n;

  def shorten(path):
    return os.path.sep.join(path.split(os.path.sep)[-level:]);

  if not isinstance(left, list): left = [left];
  if not isinstance(right, list): right = [right];
  for _ in left + right:
    if not os.path.isfile(_) and verbose:
      print(f"intersect(): invalid input {_}; exit.",
            file = sys.stderr);
      return None;
```

```
counts = {"left": 0, "right": 0, "both": 0,
          "l": [shorten(_) for _ in left],
          "r": [shorten(_) for _ in right],
          "m": 0, "n": 0};
left = connect(left);
right = connect(right);
while len(left) and len(right):
  #
  # _fix_me_ should use a genuine priority queue
  #
  left.sort(key = itemgetter(0));
  right.sort(key = itemgetter(0));
  l = left[0]; r = right[0];
  if l[0] == r[0]:
    counts["m"] += 1;
    lkey, lcount, l = left.pop(0);
    lcount += match(left, lkey, counts);
    rkey, rcount, r = right.pop(0);
    rcount += match(right, rkey, counts);
    j = min(lcount, rcount);
    counts["both"] += j;
    counts["left"] += lcount - j;
    counts["right"] += rcount - j;
    lkey, lcount, l = parse(l);
    if lkey is None: counts["n"] += l["n"];
    else: left.append((lkey, lcount, l));
    rkey, rcount, r = parse(r);
    if rkey is None: counts["n"] += r["n"];
    else: right.append((rkey, rcount, r));
  elif l[0] < r[0]:
    lkey, lcount, l = left.pop(0);
    counts["left"] += lcount;
    lkey, lcount, l = parse(l);
    if lkey is None: counts["n"] += l["n"];
    else: left.append((lkey, lcount, l));
  else:
    rkey, rcount, r = right.pop(0);
    counts["right"] += rcount;
    rkey, rcount, r = parse(r);
    if rkey is None: counts["n"] += r["n"];
    else: right.append((rkey, rcount, r));

counts["left"] += drain(left, counts);
counts["right"] += drain(right, counts);

if verbose:
```

```
    print("intersect: {} / {}.".format(counts["l"], counts["r"]), file = log);
    l = counts["left"]; r = counts["right"]; b = counts["both"];
    print("intersect(): {} shared records (of {} + {} = {}); {:.2f}% and {:.2f}% overlap."
          "".format(b, l + b, r + b, l + r + b, b / (l + b) * 100, b / (r + b) * 100),
          file = sys.stderr);
  return counts;

def intersect_directory(path, pattern = "CC-MAIN-*",
                        key = "signatures", slices = 10, cores = 8):

  all = sorted(glob.glob(os.path.join(path, pattern)));
  sample = [];
  suffix = "." + key + ".zst";
  i = 0;
  while i < len(all):
    sample.append(os.path.join(all[round(i)], suffix));
    i += len(all) / (slices - 1);
  sample.append(os.path.join(all[-1], suffix));

  with mp.Pool(cores) as pool:
    results = pool.starmap(intersect,
                           ((sample[i], sample[j], False)
                            for i in range(0, slices)
                            for j in range(i + 1, slices)));

  for counts in results:
    print("intersect: {} / {}.".format(counts["l"], counts["r"]));
    l = counts["left"]; r = counts["right"]; b = counts["both"];
    print("intersect(): {} shared records (of {} + {} = {}); {:.2f}% and {:.2f}% overlap."
          "".format(b, l + b, r + b, l + r + b, b / (l + b) * 100, b / (r + b) * 100));

  return counts;

def inspect(inputs):

  if not isinstance(inputs, list): inputs = [inputs];
  for _ in inputs:
    if not os.path.isfile(_):
      print(f"inspect(): invalid input {_}; exit.",
            file = sys.stderr);
      return None;
  inputs = connect(inputs);
  counts = {"unique": 0, "repeat": 0, "n": 0};
  while len(inputs):
    inputs.sort(key = itemgetter(0));
    key, count, entry = inputs.pop(0);
```

```
    count += match(inputs, key, counts);
    counts["unique"] += 1
    counts["repeat"] += count - 1;
    key, count, entry = parse(entry);
    if key is None: counts["n"] += entry["n"];
    else: inputs.append((key, count, entry));

  r = counts["repeat"]; u = counts["unique"];
  print("inspect(): {} repeated records (of {}); {:.2f}% self-overlap."
        "".format(r, u + r, r / (u + r) * 100));
  return counts;

def deliverable():
  path = "/appl/local/openeurollm/training/catalogue";
  d = ["fineweb/2.1.0/data", "hplt/2.0/cleaned"];
  for k in ["domains", "urls", "signatures"]:
    f = []; h = [];
    for l in ["nob", "nno"]:
      f.append(os.path.join(path, d[0], l + "_Latn", "." + k + ".zst"));
      h.append(os.path.join(path, d[1], l + "_Latn", "." + k + ".zst"));
    print(f"{d[0]} {k}:");
    inspect(f)
    print(f"{d[1]} {k}:");
    inspect(h)
    intersect(f, h, verbose = True)
  d = "madlad-400/1.0/clean";
  m = os.path.join(path, d, "nor_Latn", "." + k + ".zst");
  print(f"{d} {k}:");
  inspect(m)
  intersect(f, m, verbose = True)
  intersect(h, m, verbose = True);
  intersect_directory(os.path.join(path, "fineweb/1.4.0/data"), pattern = "CC-MAIN-*");
  inspect(os.path.join(path, "fineweb/1.4.0/data/CC-MAIN/2013/.signatures.zst"));
```

Bibliography

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.

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. URL https://arxiv.org/abs/2101.00027.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NeurIPS '24, Red Hook, NY, USA, 2025a. Curran Associates Inc. ISBN 9798331314385.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. CulturaX: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.377/.

Sneha Kudugunta, Isaac Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. MADLAD-400: A multilingual and document-level large audited dataset. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Ona de Gibert, Graeme Nail, Nikolay Arefyev, Marta Bañón, Jelmer van der Linde, Shaoxiong Ji, Jaume Zaragoza-Bernabeu, Mikko Aulamo, Gema Ramírez-Sánchez, Andrey Kutuzov, Sampo Pyysalo, Stephan Oepen, and Jörg Tiedemann. A new massive multilingual dataset for high-performance language technologies. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 1116–1128, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.100/.

Laurie Burchell, Ona De Gibert Bonet, Nikolay Arefyev, Mikko Aulamo, Marta Bañón, Pinzhen Chen, Mariia Fedorova, Liane Guillou, Barry Haddow, Jan Hajič, Jindřich Helcl, Erik Henriksson, Mateusz Klimaszewski, Ville Komulainen, Andrey Kutuzov, Joona Kytöniemi, Veronika Laippala, Petter Mæhlum, Bhavitvya Malik, Farrokh Mehryary, Vladislav Mikhailov, Nikita Moghe, Amanda Myntti, Dayyán O'Brien, Stephan Oepen, Proyag Pal, Jousia Piha, Sampo Pyysalo, Gema Ramírez-Sánchez, David Samuel, Pavel Stepachev, Jörg Tiedemann, Dušan Variš, Tereza Vojtěchová, and Jaume Zaragoza-Bernabeu. An expanded massive multilingual dataset for high-

performance language technologies (HPLT). In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 17452–17485, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.854/.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. FineWeb2: One pipeline to scale them all – adapting pre-training data processing to every language, 2025b. URL https://arxiv.org/abs/2506.20920.

Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-CC: Transforming Common Crawl into a refined long-horizon pretraining dataset. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2459–2475, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. URL https://aclanthology.org/2025.acl-long.123/.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dav, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NeurIPS '24, Red Hook, NY, USA, 2025. Curran Associates Inc. ISBN 9798331314385.

Nikhil Kandpal, Brian Lester, Colin Raffel, Sebastian Majstorovic, Stella Biderman, Baber Abbasi, Luca Soldaini, Enrico Shippole, A. Feder Cooper, Aviya Skowron, John Kirchenbauer, Shayne Longpre, Lintang Sutawika, Alon Albalak, Zhenlin Xu, Guilherme Penedo, Loubna Ben Allal, Elie Bakouch, John David Pressman, Honglu Fan, Dashiell Stander, Guangyu Song, Aaron Gokaslan, Tom Goldstein, Brian R. Bartoldson, Bhavya Kailkhura, and Tyler Murray. The Common Pile v0.1: An 8tb dataset of public domain and openly licensed text, 2025. URL https://arxiv.org/abs/2506.05209.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.577. URL https://aclanthology.org/2022.acl-long.577/.

Hao Xu, Jiacheng Liu, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Infini-gram mini: Exact n-gram search at the internet scale with fm-index, 2025. URL https://arxiv.org/abs/2506.12229.