

Task-Agnostic Web Document Annotation with LLM-Generated Descriptors

Otto Tarkka University of Turku TurkuNLP ohitar@utu.fi	Erik Henriksson University of Turku TurkuNLP erik.henriksson@utu.fi	Jenna Kanerva University of Turku TurkuNLP jmnybl@utu.fi	Filip Ginter University of Turku TurkuNLP figint@utu.fi
--	---	--	---

Abstract

Web-scale text corpora provide NLP researchers and corpus linguists with access to vast quantities of language data, but efficiently identifying the subsets of a large corpus that are relevant to a particular research aim remains a significant challenge. Existing approaches, including keyword search and task-specific classifiers, are often either too restrictive or depend on costly retraining for each new research question. We explore and evaluate the feasibility of a task-agnostic, LLM-driven descriptor schema for annotating web documents, using LLMs to generate natural-language labels that capture stylistic, tonal, topical, and structural properties of texts. Unlike prior work that emphasizes broad categorization or quality filtering, our approach yields fine-grained and interpretable descriptors designed to support flexible downstream applications. We evaluate the approach from multiple angles and show that descriptor-based annotation can enable flexible document retrieval while also exposing challenges related to schema harmonization and descriptor specificity.

1 Introduction

Natural language corpora have grown rapidly from the 1-million-word Brown corpus built in the 1960s to multi-trillion word corpora today. Large web corpora provide unprecedented insight into language use on the internet and are also used to train state-of-the-art Large Language Models (LLMs). However, large corpora are often only the starting point for a research project and are rarely used in their entirety. Corpus linguists typically only focus on a small subset of documents that are relevant to their research questions. Even for LLM training, requiring trillions of tokens, identifying the highest quality documents can be more important than focusing on quantity (Su et al., 2025; Lozhkov et al., 2024; Li et al., 2024; Diao et al., 2025).

To facilitate broader adoption of these vast collections in research, the data need rich annotations

and metadata. In this paper, we build and evaluate a data-driven task-agnostic approach for generating open-ended document annotations. We leverage LLMs to generate a comprehensive document descriptor schema with more than 800,000 unique labels that can be used to create subsets of data for many purposes with relative ease. We present a proof-of-concept implementation of the approach and the steps taken in developing it and evaluate the descriptors for accuracy, aspect coverage, and usability. We release our code and the descriptor schema, which can be applied to any English corpus of web documents, and a ready-to-use, 10-billion-token descriptor-annotated corpus.

2 Background and related work

The idea of using the internet as a source of large-scale language data began at the start of the millennium with the Web-as-Corpus approach (Kilgariff and Grefenstette, 2003). The value of metadata in large web corpora has also long been recognized (e.g., Barbaresi 2016) and, recently, LLMs have emerged as tools for annotating and analyzing text data at scale (Tan et al., 2024; Yadav et al., 2024; Chen et al., 2024a,b). ChatGPT and other state-of-the-art LLMs already outperform human coders in many annotation tasks (e.g., Bermejo et al. (2024); Gilardi et al. (2023); Törnberg (2024)). For example, Maragheh et al. (2023) use an LLM to generate thematic keywords for English news and e-commerce documents while Zhang and Takada (2025) use GPT-4o in an active learning setting, where documents are labeled with the LLM for classification tasks.

Identifying valuable documents for LLM training from large web corpora has become increasingly important (Penedo et al., 2024; Chen et al., 2024a; Su et al., 2025; Li et al., 2024). To this end, Wettig et al. (2025) introduce WebOrganizer, two parallel taxonomies that characterize the topic and format of web documents. They use Llama models

to annotate documents and train smaller classifiers for subsequent large-scale annotation. [Lozhkov et al. \(2024\)](#) use an LLM to annotate web documents for their educational value and train a classifier to produce FineWeb-Edu. [Idahl et al. \(2026\)](#) use an LLM to annotate web documents along 18 quality axes such as completeness, structure, and information density.

All of the approaches above follow the same basic steps: (1) Annotate data. (2) Train a classifier on the annotated data. (3) Run inference on the full dataset. This requires considerable compute effort, as well as extensive human labor if the training data is manually annotated. These approaches rely on manually crafted, task-specific annotation schemas and categorize documents into a relatively small number of broadly defined categories with coarse labels. While this approach works, it limits the usability of the annotations to applications directly supported by the annotation schema. For instance, if one were to seek to retrieve “sarcastic” documents in the presently available large web corpora, no metadata in these corpora allow that query to be made. Consequently, the cycle of annotation, training, and inference on a large corpus must be repeated for this and every other new task not directly supported by the existing annotation schemas.

While training new classifiers is feasible for major undertakings, a myriad of smaller tasks and niche research questions remain unexplored due to this constraint. Therefore, in this paper we explore an approach where each document is given a task-agnostic annotation that is detailed enough to support a wide variety of tasks. In this bottom-up approach, an LLM generates a descriptor schema spanning hundreds of thousands of unique labels instead of relying on a few predetermined categories. Crucially, the data is annotated just once, after which the descriptors can be used with little cost and effort either on their own or as a first-pass filter. We propose our approach as a complement to, not a replacement for, traditional classification schemas. In our view, the more metadata large corpora have, the more accessible they are for all corpus users.

This philosophy of scalable, open-ended annotation shares conceptual ground with prior work, though with key distinctions. For example, TopicGPT ([Pham et al., 2023](#)), similarly employs an LLM to generate topic labels for documents. How-

descriptor	
tag	definition
formal tone and language	The formal tone and language used convey credibility, authority, and expertise, making it suitable for professional or official contexts.

Figure 1: Example of a descriptor, consisting of a tag and a definition.

ever, our method differs in several key respects: (1) TopicGPT assigns only topic labels, whereas our descriptors capture additional document aspects (see Section 4.2); (2) the scale of labels in TopicGPT is limited to dozens, rather than the hundreds of thousands in our schema; and (3) TopicGPT relies on predefined seed labels to guide annotation. [Lu et al. \(2024\)](#) also employ an LLM for open-ended annotation, but their work focuses on tagging LLM instructions to facilitate efficient model fine-tuning rather than general-purpose document description.

3 Method

Our objective is to assign each document a number of *descriptors*, where each descriptor captures some aspect of the document. Every descriptor consists of two parts: a unique *tag* and a *definition* (see Figure 1). The tag is a short phrase (usually 1–3 words) that holds the main meaning, while the definition expands on it by providing additional context and specificity. The definitions help distinguish between otherwise ambiguous or opaque tags, and add detail and length for better embeddings, which is useful for downstream tasks.

3.1 The descriptor schema

As the basis of our descriptor schema, we use a sample of 500,000 documents from FineWeb, a collection of relatively high-quality English documents ([Penedo et al., 2024](#)). This sample is selected to ensure comprehensive coverage of the diverse text types, topics, and formats present on the English web. We label the data using Llama-3.3-70B-Instruct ([Grattafiori et al., 2024](#)). In initial experiments, we found that this model generates high-quality descriptors while being computationally manageable at our data sizes.¹

The annotation is carried out in an open-ended fashion, with descriptors created dynamically dur-

¹While newer models have become available since our initial runs, re-running all experiments is unlikely to produce radically different results to justify the resource cost.

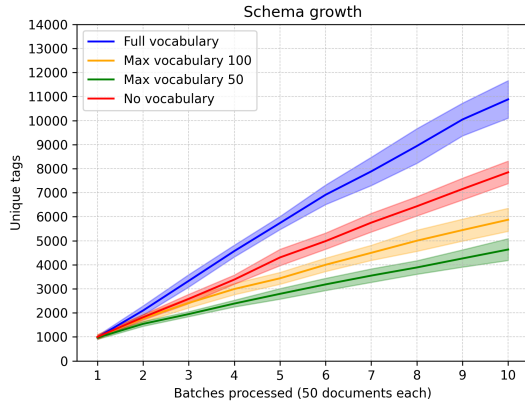


Figure 2: Rate of tag schema growth with different maximum vocabularies (average of 5 runs, batch size 50). Vocabulary size refers to the number of existing tags given in the prompt.

ing the annotation process. Since each document is labeled independently, a major methodological challenge is ensuring that the descriptors are detailed enough to cover all aspects of the document while avoiding semantic duplication, where the same phenomena are annotated with differently worded descriptors. The most straightforward approach would be to inform the model about descriptors assigned in previous data batches in the prompt and instruct it to introduce new descriptors only when necessary. Ideally, the annotation schema would grow quickly at first and then plateau, as most of the required labels would already exist. This approach would generate a comprehensive labeling schema in a single pass through the data. To examine this strategy, we conducted a preliminary experiment using tags alone, on the assumption that the model would find it easier to track existing tags than full descriptors. Figure 2 shows the growth of the tag vocabulary under three prompting conditions: providing the full, growing tag vocabulary, providing no vocabulary, or providing only the top N most common prior tags.

This experiment indicated that the model struggles to adhere to the given schema. We observed substantial levels of synonymous tags in the model output even when the model was informed of suitable existing tags. Moreover, contrary to expectations, providing a larger vocabulary accelerated the rate at which new tags were generated. Prompt tuning alone was insufficient to mitigate this issue.

Since generating the schema and labeling the documents in a single pass was found to be unfeasible, an alternative approach is to first generate de-

scriptors freely and subsequently refine the schema by merging duplicates and synonyms. Our first attempt was to cluster and merge new descriptors with the existing schema after each new batch of data. While this slowed the growth of the descriptor schema, descriptors that were lexically similar but semantically opposed, such as “Formal” and “Informal”, tended to get merged. To avoid this, the LLM was tasked with deciding whether and how to merge a cluster. While this approach worked better, throughput slowed drastically, preventing large-scale annotation. Hence, merging was implemented in two separate post-processing steps.

The final pipeline consists of three steps: (1) Generation: the initial descriptors are generated for each document independently; (2) Disambiguation: descriptors with duplicate tags are either merged, with their definitions revised accordingly, or renamed so that each unique descriptor has a unique tag; (3) Synonym merging: synonymous or near-synonymous descriptors are merged. These steps together produce a descriptor schema where each unique tag has exactly one definition and the prevalence of synonymous or near-synonymous descriptors is minimal.

To preserve space, the descriptions of the steps in the following sections are abbreviated. For a more comprehensive description and the full prompts used, refer to Appendices A and D.

3.1.1 Generation step

The generation step consists of three stages: generation, rewriting, and revision. First, the model is given one document and is prompted to “[d]escribe the document in aspects including, but not limited to, style, tone, genre, topic, domain, language, quality, sentiment, etc.” For each tag, the model is also prompted to generate a definition, which should “explain its meaning and relevance to the document in 2 to 3 sentences”. The generated descriptors should be detailed enough to support approximate reconstruction of the original content from the descriptors alone. However, to avoid mixing highly specific details with more general features, the model is prompted to place minute details under *specifics*. These are used to evaluate the model’s ability to reconstruct the document but are not used thereafter.

To evaluate how well the meaning of the document is captured in the descriptors, we give the descriptors generated in the previous stage back to the model and ask it to rewrite the corresponding

document. The rationale behind this step is that a set of descriptors accurately reflecting as many aspects of a document as possible should, in principle, serve as a robust blueprint for rewriting it. If the generated text closely resembles the original, this suggests that the descriptors have successfully preserved its meaning.

Since the LLM might not generate high-quality descriptors in its first attempt, the model is asked to evaluate the quality of the rewrite by comparing it to the original and to revise the descriptors to produce a more faithful rewrite next time. The revision stage is followed by another rewrite, now using the revised descriptors, and then another revision. Although this process could be repeated arbitrarily many times, we stop after three sets of descriptors and rewrites to preserve computational resources.

To identify the set of descriptors that best captured the meaning of the original document, we embed the original document and each of its rewrites using the Stella-en-400M-v5 model (Zhang et al., 2025) and calculate their cosine similarity. The closer a rewrite is to the original document, the better the descriptors have summarized the document and thus, the better their quality. Our experiments confirm that rewrite similarity increases with more descriptor revisions (Figure 3).

Only the best descriptors for each document are kept. Labeling 500,000 documents results in nearly 7.5 million unique descriptors. Each document gets an average of 15 descriptors, and some of the most common tags are “formal tone”, “conversational tone”, and “technical”.

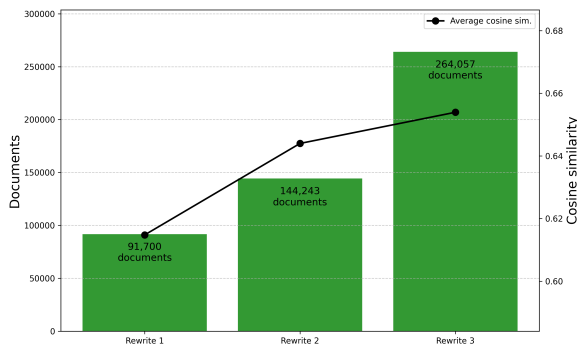


Figure 3: Number of documents whose best rewrite was generated on the first, second, and third attempts. The best rewrite was most often generated on the third attempt, although the average increase in cosine similarity was modest.

Step	Descriptors
Generation	7,433,036
Disambiguation	1,001,994
Synonym merging	800,488

Table 1: Number of unique descriptors after each step in the pipeline.

3.1.2 Disambiguation step

In the disambiguation step, we normalize all tags to lowercase and group together all definitions for a given tag. The vast majority of tags have only one definition, but some have up to thousands of different definitions. The groups with more than one definition are given to the LLM, which is prompted to either merge (near-)synonymous entries and provide one comprehensive descriptor or split the group into several descriptors, each consisting of a locally unique tag and definition.

3.1.3 Synonym merging step

After disambiguation, the schema still contains many (near-)synonymous descriptors because only exact tag duplicates were grouped together in the previous step. For instance, tags such as “formal tone” and “formal language” are still separate at this point. In the synonym merging step, each descriptor is first embedded using the Stella model, and mutually nearest neighbors are provided to the LLM. The model either accepts them as synonyms and chooses one to represent both or rejects their synonymy and keeps them as separate entries. After each iteration, new mutually nearest neighbors are identified and provided to the model until none are left. After synonym merging, the descriptor schema has condensed from nearly 7.5 million to around 0.8 million unique descriptors, where each tag is unique and has only one definition (Table 1).

3.2 Harmonization

The previous three steps, when carried out on a sufficiently large number of documents, will result in a schema of descriptors capable of describing a broad variety of documents. Any new documents can thereafter be annotated in this schema, without further updates. For new documents, we first apply the descriptor generation step (Section 3.1.1). This produces document-specific descriptors that do not necessarily correspond to the schema. We refer to these independently generated descriptors as *raw* descriptors throughout the remainder of the paper.

Next, these raw descriptors are mapped onto the

existing schema. We refer to the resulting schema-mapped descriptors as *harmonized* descriptors. For each raw descriptor, we retrieve the 10 most similar descriptors from the schema² and use the Llama model to select the best match among them. If the model rejects all 10 candidates, the descriptor is dropped, although in our experiments, this only happens to less than 0.5% of the descriptors. In the end, each document has a set of raw descriptors and a similar but schema-adhering set of harmonized descriptors.

While the unharmonized raw descriptors can be used as-is without harmonization, the harmonization step achieves two goals: First, it ensures the descriptors adhere to the frozen descriptor schema, which makes descriptor annotations between datasets and possibly different annotator models comparable and compatible with the same downstream processing methods. For example, examining the label distribution of two descriptor annotated corpora could reveal biases in their content. This comparison would be much more difficult to do without harmonized descriptors. Second, harmonization limits the descriptor space by reducing the number of unique descriptors. This is desirable because it makes the descriptors easier to use and greatly reduces the number of singleton descriptors (i.e., descriptors that appear in a single document), which leads to better recall in retrieval and classification tasks (see Sections 4.3 and 4.4).

4 Evaluation

We evaluate the descriptors on a sample of 100,000 FineWeb documents that were not used to derive the descriptor schema. We annotate the test documents with raw descriptors and then harmonize them and evaluate both raw and harmonized descriptors. On average, each document in the test set received 15 descriptors. Raw descriptors follow a long-tailed distribution, where the majority appear in only a single document, while the average harmonized descriptor appears in close to 5 documents (Figure 4).

For much of the evaluation, we rely on an LLM-as-judge approach, as human annotation would be prohibitively expensive. We use Qwen3-Next-80B-A3B-Instruct (Qwen Team, 2025) as the judge model. This model was chosen due to its strong performance and computational efficiency. It is

²Based on cosine similarity of their embeddings calculated using the Stella model.

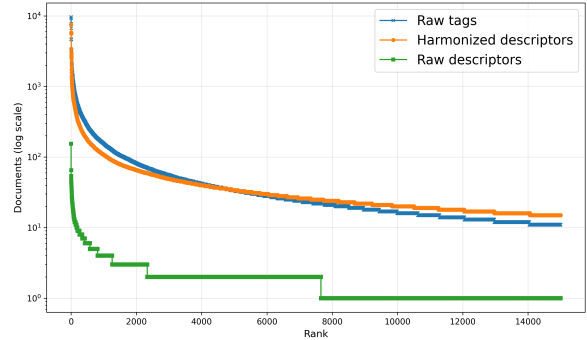


Figure 4: The number of documents in which the 15,000 most common tags and descriptors occur in the harmonized and raw settings. Harmonized tags not shown since their counts equal harmonized descriptors.

from a different model family than the annotator model, which mitigates same-model bias. A small scale human annotation test is conducted to verify the reliability of the model (Section 4.1).

4.1 Evaluation 1: Descriptor accuracy

For the descriptors to be useful, they need to be accurate. While we can expect raw descriptors to be mostly accurate, since research has indicated that LLMs perform well as data annotators, some loss in accuracy is difficult to avoid during harmonization when a raw descriptor is mapped to its (near-)synonym from the schema.

We evaluate the accuracy of the descriptors on a sample of 5,000 documents from the test data. The judge model is given a document with its descriptors and is prompted to evaluate how accurately the descriptors describe the document on a five-point scale. The results are summarized in Table 2. Almost 90% of the raw descriptors are judged to be accurate, while only approximately 5% are inaccurate or mostly inaccurate. As expected, the harmonized descriptors are less accurate. Still, nearly 80% are judged to be at least partially accurate. This result shows that while the model does the initial descriptor annotation with high accuracy, the process of harmonizing the descriptors with an existing schema requires further development in future work.

We also have two human annotators annotate a sample of 20 documents and their descriptors along the same 5-point-scale as the LLM judge. The inter-annotator agreement (weighted Cohen’s Kappa) between the two human annotators is 0.43 for raw descriptors and 0.58 for harmonized descriptors. Agreement between annotators and the LLM judge

Model judgment	Raw	Harmonized
Accurate	88.67%	61.81%
Mostly accurate	0.83%	2.90%
Partially accurate	5.58%	14.61%
Mostly inaccurate	0.55%	1.51%
Inaccurate	4.36%	19.16%

Table 2: Descriptor accuracy as determined by LLM judge.

ranges from 0.37 to 0.49 for raw descriptors and 0.43 to 0.49 for harmonized descriptors. These scores indicate moderate agreement between human annotators and the judge model. The modest kappa is partially explained by the skewed label distribution and resulting high chance agreement (see confusion matrices in Figures 6 and 7 in Appendix E). Most disagreements are adjacent (e.g., Accurate vs. Mostly Accurate), suggesting agreement on the general accuracy of descriptors but disagreement in fine-grained assessments. While the annotations are noisy, they provide reasonable confidence both in the accuracy of the descriptors and in the performance of the LLM judge model.

4.2 Evaluation 2: Document aspect coverage

A primary motivation for our method is to create metadata that can be used to create subsets of data for tasks that are not known when the annotation schema is created. Therefore, we analyze the aspects the descriptors cover and how they are distributed. We take a sample of roughly 5,000 documents, containing almost 75,000 descriptors, and use the same LLM judge model to categorize each descriptor into one or more aspect categories (see Appendix F for detailed description). Then, for each document, we measure if a given aspect category is present in at least one of its descriptors.

The results in Table 3 show that the most common aspect described by descriptors is Content & Subject Matter, followed by Style & Tone and Structure & Format. In terms of document coverage, we find that almost all documents have at least one descriptor relating to Content & Subject Matter. Coverage of other aspects is less common but still 69% of documents cover at least 4 aspects and 35% cover at least 5. The results for harmonized descriptors are very similar. In sum, document topics are the most covered aspect but other aspects are covered as well, including aspects not directly mentioned in the initial generation prompt.

4.3 Evaluation 3: Comparison to existing annotation schemes

In this section, we compare descriptors to two existing web document annotation schemes to assess how well the descriptors capture the information contained in these task-specific schemes. We apply the WebOrganizer classifiers, which assign each document with a format and a topic label (Wettig et al., 2025), to our test data. We also use the educational labels from FineWeb-Edu (Lozhkov et al., 2024). While these classification schemes provide a useful benchmark, machine-generated labels may contain inaccuracies. Subjectivity further complicates evaluation: we test this by using the LLM judge to evaluate whether the labels derived from the WebOrganizer classifiers are accurate. The LLM judge agrees with the classifier labels less than 60% of the time. Additionally, the descriptors offer a much more granular representation of content, with an average of 15 descriptors per document. With these limitations in mind, while we use WebOrganizer and FineWeb-Edu labels as comparative reference points, our results should be interpreted as indicative of broader trends rather than definitive ground truth.

First, we compare descriptors to WebOrganizer labels by providing the LLM judge with all the descriptors of a given document and asking which format and topic labels can be reasonably inferred from them. We find that 77% of topic labels could be inferred from both raw and harmonized descriptors, while inference rates for format labels were 57% (raw) and 53% (harmonized). This indicates that the descriptors often contain information about

Aspect	Descriptor coverage	Document coverage
Content & Subject Matter	43,284 (58%)	4,898 (99%)
Style & Tone	15,034 (20%)	4,304 (87%)
Structure & Format	7,236 (10%)	2,673 (54%)
Quality & Trustworthiness	3,837 (5%)	2,619 (53%)
Audience & Engagement	3,297 (4%)	2,600 (53%)
Ethical & Legal	1,164 (1%)	835 (17%)
Linguistic Features	1,103 (1%)	1,881 (38%)
Other	80 (<1%)	132 (3%)

Table 3: Aspect categories of raw descriptors according to the LLM judge. Descriptor coverage refers to the number of descriptors in the 5,000 document sample that cover the corresponding aspect. One descriptor may belong to more than one category. Document coverage refers to the number of documents that have at least one descriptor that covers the corresponding aspect.

a document’s topic while information about its format is rarer, while still often present, which supports the results in Section 4.2.

Next, we train logistic regression classifiers using descriptors as features to predict topic, format, and educational labels and compare them against classifiers trained with TF-IDF features. This tests how much information relevant to predicting these labels is carried in the descriptors and how accessible this information is to a linear classifier compared with information encoded in *n*-grams. We split the test data into 80/20 training and test sets and target the binary FineWeb-Edu labels and the WebOrganizer topic and format taxonomies, each with 24 classes. For each task, we train a one-vs-rest logistic regression model.

We use the *tags* (not full descriptors) as features, encoded as a binary presence vector. In the harmonized setting, each tag has one definition, making harmonized tags and descriptors equivalent. In the raw setting, however, most full descriptors are unique to a single document, providing little basis for learning. Tags, in contrast, recur across documents and can thus serve as meaningful predictive features. The TF-IDF reference models are trained on 1–2-grams from full texts. The full per-class retrieval results appear in Appendix G (Tables 7–12).

The raw descriptors perform on par with TF-IDF capped at 50,000 features (Table 6) on macro-F1 on the educational (0.76 vs. 0.75) and format labels (0.48 vs. 0.49), but fall behind on topic (0.65 vs. 0.73). Even though our other evaluations show that topics are well represented in the descriptor space, their large number hints at very fine topic distinctions, which lead to sparse features and low performance. Raw descriptors outperform harmonized ones on all three tasks, reflecting the accuracy loss from harmonization.

The strength of descriptors as features is in their interpretability: for example, most of the highest-weighted features for non-educational documents are interpretable phrases (“product description” and “international travel guide”), whereas TF-IDF’s are opaque tokens (“me”, “my”, and “2018”) (Table 13, Appendix G).

We also test the applicability of the descriptor-trained models as cheap filters by ranking the documents by the model’s score for each class to find the fraction of the test corpus that must be scanned to reach 80% and 20% recall. To reach 80% recall on topic, the descriptors require scanning 11.6–11.8%

of the corpus against 7.2% for TF-IDF, and on format 22–24% against 16%. The gap narrows considerably at the top of the ranking: at 20% recall the descriptors scan nearly the same fraction as TF-IDF (1.3% vs. 1.1% on topic, 2.6–2.7% vs. 2.2% on format). Performance also varies by class: the descriptors are strongest on semantically coherent classes such as Adult, Religion, and Legal Notices, and weakest on classes defined by surface form or boilerplate, such as FAQ and Content Listing. Overall, the results in this section suggest that many of the distinctions made by existing purpose-built web annotation schemes are also reflected in the descriptors to some extent, despite the descriptors not being explicitly designed for them.

4.4 Evaluation 4: Retrieving documents without a classifier

One of our central goals is to enable fast and simple retrieval from a large corpus. To test this, we perform embedding-based similarity search for WebOrganizer topic and format labels from our test data. We embed each label together with its definition (Wettig et al., 2025, pp. 17–18) as a query. Additionally, we query for “sarcastic” documents as an example of a text type that could be of interest to researchers but is not covered by popular annotation schemes and is difficult to retrieve with a simple keyword or regular expression matching.

The evaluation is done in four steps: (1) Format each query as a descriptor (tag + definition). (2) Retrieve the nearest test set descriptors via Stella embeddings and a Faiss index (Douze et al., 2026), using a fixed squared Euclidean distance threshold of 320, which we experimentally find retrieves mostly plausible results without excluding too many potentially relevant hits. (3) Apply the LLM judge to filter retrieved descriptors for query relevance. (4) Use the resulting set of query-matched descriptors to retrieve the corresponding documents.

Using labels given by the WebOrganizer format and topic classifiers as the ground truth, we find that some labels are retrieved more accurately (e.g., Politics and Personal Blog) than others (e.g., Spam & Ads), while topic labels perform better than format labels (Table 4, full results in Appendix H). While raw descriptors have slightly higher precision, harmonized descriptors have better recall since a raw descriptor typically only retrieves one document, while one harmonized descriptor might appear in hundreds of documents.

In light of the issues with the WebOrganizer labels discussed in 4.3, we also evaluate the retrieval results with the LLM judge. The judge evaluates whether the documents associated with the approved descriptors correspond to the query. The judge is given three response options: “Yes”, “No”, and “Partial”, indicating full, no, and partial correspondence, respectively. The “Partial” option is given to reflect the more granular nature of the descriptors compared to the WebOrganizer labels, although the LLM judge rarely uses this option. This evaluation shows a higher rate of success for retrieved documents, with 56% and 75% full or partial correspondence for formats and topics, respectively, retrieved with harmonized descriptors. For formats and topics retrieved with raw descriptors, the model judges full or partial correspondence of 67% and 83%, respectively.

	Raw		Harmonized	
	Prec.	Rec.	Prec.	Rec.
formats	0.35	0.25	0.30	0.29
topics	0.64	0.35	0.57	0.42

Table 4: Average precision and recall for topic and format queries

The sarcasm query retrieved 5,459 documents with harmonized descriptors and 3,821 documents with raw descriptors. Since we do not have labels indicating sarcasm, we evaluate the retrieval of “sarcastic” documents only using the LLM judge. Out of the documents retrieved with harmonized descriptors, the judge model classified 74% as full or partial matches. For documents retrieved with raw descriptors, the acceptance rate is 85%.

As Figure 5 shows, descriptor acceptance generally declines with decreasing descriptor–query similarity. However, documents retrieved through LLM-approved descriptors are accepted at high rates, indicating that the approved descriptors are reliable indicators of sarcasm. The lower and more variable document-level acceptance rates for harmonized descriptors reflect their broader coverage: a single harmonized descriptor often retrieves many documents, some of which may not be relevant to the query.

This section has presented and evaluated a simple yet effective method to retrieve documents that would otherwise be difficult to identify, requiring only a user-provided query. While exhaustive recall is not guaranteed and the retrieved documents may

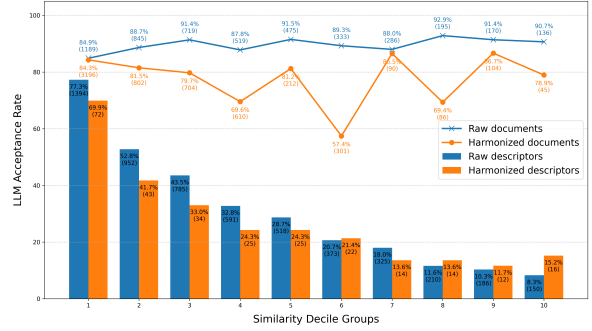


Figure 5: LLM acceptance rates for descriptors and documents retrieved by the sarcasm query, by descriptor–query similarity decile. Bars show the share of retrieved descriptors accepted by the LLM judge; lines show the share of associated documents accepted by the judge. Absolute counts are in parentheses. Document counts may exceed the number of unique retrieved documents because a document can be retrieved multiple times by different descriptors.

include false positives, this method may serve as a fast way to obtain a substantial subset of relevant documents or as a preliminary filtering step to reduce the search space for more resource-intensive classification methods.

5 Discussion and conclusions

In this paper, we have explored a novel approach for enriching web corpora with metadata that is both human-readable and searchable while remaining sufficiently detailed to support a wide range of tasks without the need to specify them *a priori*. The main motivation of this study is to enable the categorization and retrieval of task-specific subsets of large web corpora without having to apply the resource-intensive annotate–train–classify approach separately to each new task. The effort of assigning the descriptors needs to be spent only once, after which they can be reused across applications. In our evaluation of the descriptor schema, we also implicitly evaluate the capability of LLMs for this task.

Our first finding is that LLMs can assign detailed, human-readable, and useful descriptors to documents, even without a predefined schema. First, we show that the descriptors accurately describe the documents (Section 4.1), although the current harmonization approach leads to a loss in accuracy. Second, we show that the descriptors cover many different aspects (Section 4.2). While topic is clearly the most covered aspect, style, tone, format, and quality are also represented, among others.

Third, we show that the descriptor schema contains much of the same information as two existing web document annotation schemes (Section 4.3). Finally, we show that a simple and fast query procedure can retrieve a non-trivial number of documents matching topics, formats, and even sarcasm; phenomena that would be difficult to identify in large corpora without a fine-tuned classifier (Section 4.4).

Our second finding is that while LLMs can generate detailed and useful descriptors, they struggle with systematicity across large document sets. Attempts to devise the descriptor schema and annotate the documents in a single pass were unsuccessful. We were able to alleviate but not fully solve the problem by disambiguating and merging the descriptors post hoc using a combination of embedding similarity and LLM judgments. Additionally, the high granularity of descriptors, while advantageous for the precise retrieval, complicates the identification of broad categories, such as “sports” or “news”. Future work should explore new ways to enforce schema consistency and hierarchical descriptor structures for adjustable granularity and develop methods to efficiently use larger schemas. Some of our evaluation strategies, especially those in Sections 4.3 and 4.4 could be developed further into practical use cases.

To conclude, we strive to create rich, human-readable metadata for large corpora, produced only once and applicable to any number of tasks to complement existing annotation schemes. While the work presented in this paper is only one step in this direction and much work remains to be done, we envision the descriptors being used as a first-pass filter before more resource intensive classification or as a standalone tool for researchers lacking the means to train custom models. With further method development, such metadata could democratize access to large corpora for new user groups and increase the efficiency of current practices.

The pipeline and evaluation code are made available.³ We also release a descriptor-annotated 10-billion-token sample of FineWeb, the full descriptor schema, the schema development and test sets as well as disambiguation and synonym merging traces.⁴

³https://github.com/TurkuNLP/LLM_document_descriptors

⁴<https://huggingface.co/datasets/TurkuNLP/WebDocumentDescriptors>

Limitations

This research is limited to monolingual English documents from the FineWeb dataset, which is purposefully built to exclude low-quality documents. While the schema is built and evaluated on separate subsets of FineWeb, it is not tested on fully out-of-distribution data and may not generalize well to noisier datasets. Additionally, the findings do not yet show whether the approach generalizes to other languages or multilingual settings. Future work should therefore evaluate the method on multilingual corpora.

In the current implementation, descriptors serve as document-level annotations. Although a given descriptor may apply to only a single paragraph or sentence within a document, the system does not explicitly track which specific section it pertains to. Future methodological refinements could incorporate section-level information into these annotations.

The specific language model used for descriptor generation and harmonization can also be a limitation. Since LLM capabilities continue to improve rapidly, newer models may produce more accurate, consistent, and multilingual descriptors. The exact prompts given to the model also have a big impact on model output. The prompts Consequently, the results reported here should be understood as reflecting the performance of one specific model rather than the upper bound of the proposed method.

Finally, although applying existing descriptors to downstream tasks is relatively fast and inexpensive, generating descriptors for large datasets remains computationally costly. To address this limitation, future work could explore smaller fine-tuned models that generate descriptors more efficiently while preserving descriptor quality. We plan to train and release such a model in future work.

Ethical considerations

This research relies on generative LLMs, which are known to produce errors in their output. Researchers should take care to evaluate the descriptor schema themselves before applying it to their research. While we use a high-quality dataset, FineWeb, as the basis of the descriptor schema, the documents might still contain biased, racist, sexist, or otherwise unreasonable or harmful content, which may be reproduced in the document descriptors. Using LLMs consumes electricity, which,

depending on the production method, can lead to greenhouse gas emissions. For this paper, processing 500,000 documents through the pipeline required an estimated 1,000 GPU hours on four AMD MI250X GPUs and all electricity came from renewable resources.

AI assistants were used to proofread sections of this paper. Some of the code used in the experiments was written with AI assistance. All AI-generated text and code were reviewed and verified by the authors.

Acknowledgments

This research has been supported by the Digital Europe Programme under grant agreement No. 101195233 (OpenEuroLLM); the Human Diversity consortium, under the Profi7 program (grant no. 352727) and the Centre of Excellence (grant no. 374221) by the Research Council of Finland; the Research Council of Finland through FINCLARIAH research infrastructure (project 358720, which has also received funding from the European Union – NextGenerationEU instrument), and the Mechanisms of Register Variation in Massively Multilingual Web-Scale Corpora (project 362459); and the GreenNLP project (project 353167) by the Research Council of Finland.

This project is funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor EACEA can be held responsible for them.

We thank CSC – IT Center for Science, Finland, for computational resources.

References

- Adrien Barbaresi. 2016. [Efficient construction of metadata-enhanced web corpora](#). In *Proceedings of the 10th Web as Corpus Workshop*, pages 7–16, Berlin. Association for Computational Linguistics.
- Vicente J. Bermejo, Andres Gago, Ramiro H. Gálvez, and Nicolás Harari. 2024. [LLMs outperform out-sourced human coders on complex textual analysis](#). SSRN Working Paper.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024a. [Alpagasus: Training a better alpaca with fewer data](#).
- Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024b. [Is a large language model a good annotator for event extraction?](#) *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17772–17780.
- Shizhe Diao, Yu Yang, Yonggan Fu, Xin Dong, Dan Su, Markus Kliegl, Zijia Chen, Peter Belcak, Yoshi Suhara, Hongxu Yin, Mostofa Patwary, Yingyan, Lin, Jan Kautz, and Pavlo Molchanov. 2025. [Nemotron-CLIMB: CLustering-based Iterative Data Mixture Bootstrapping for Language Model Pre-training](#).
- Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2026. [The Faiss Library](#). *IEEE Transactions on Big Data*, 12(2):346–361.
- Fabrizio Gildardi, Meysam Alizadeh, and Maël Kubli. 2023. [ChatGPT outperforms crowd workers for text-annotation tasks](#). *Proceedings of the National Academy of Sciences*, 120(30).
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Al-lonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yearly, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar

Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Rapparthi, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephanie Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vitor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenxin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuwei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkan Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippas Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank

Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabza, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihalescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng

- Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. [The Llama 3 Herd of Models](#).
- Maximilian Idahl, Benedikt Droste, Björn Plüster, and Jan Philipp Harries. 2026. [propella-1: Multi-Property Document Annotation for LLM Data Curation at Scale](#).
- Adam Kilgarriff and Gregory Grefenstette. 2003. Introduction to the special issue on the web as corpus. *Computational Linguistics*, 29:333–347.
- 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, Dhruva 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 Dave, Ludwig Schmidt, and Vaishaal Shankar. 2024. [DataComp-LM: In search of the next generation of training sets for language models](#).
- Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. 2024. [FineWeb-Edu: the Finest Collection of Educational Content](#).
- Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2024. [#InsTag: Instruction tagging for analyzing supervised fine-tuning of large language models](#). The Twelfth International Conference on Learning Representations.
- Reza Yousefi Maragheh, Chenhao Fang, Charan Chand Irugu, Parth Parikh, Jason Cho, Jianpeng Xu, Saranyan Sukumar, Malay Patel, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2023. [LLM-TAKE: Theme-aware keyword extraction using large language models](#). In *2023 IEEE International Conference on Big Data (BigData)*, pages 4318–4324.
- Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. [MTEB: Massive Text Embedding Benchmark](#).
- Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al. 2024. [The FineWeb](#) [datasets: Decanting the web for the finest text data at scale](#). *Advances in Neural Information Processing Systems*, 37:30811–30849.
- Chau Minh Pham, Alexander Hoyle, Simeng Sun, and Mohit Iyyer. 2023. [TopicGPT: A Prompt-based Topic Modeling Framework](#).
- Qwen Team. 2025. [Qwen3 technical report](#).
- Dan Su, Kezhi Kong, Ying Lin, Joseph Jennings, Brandon Norick, Markus Kliegl, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. 2025. [Nemotron-CC: Transforming common crawl into a refined long-horizon pretraining dataset](#).
- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansoor Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024. [Large language models for data annotation and synthesis: A survey](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.
- Petter Törnberg. 2024. [Large Language Models Outperform Expert Coders and Supervised Classifiers at Annotating Political Social Media Messages](#). *Social Science Computer Review*.
- Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. 2025. [Organize the web: Constructing domains enhances pre-training data curation](#).
- Sachin Yadav, Tejaswi Chopra, and Dominik Schlechtweg. 2024. [Towards automating text annotation: A case study on semantic proximity annotation using gpt-4](#).
- Dun Zhang, Jiacheng Li, Ziyang Zeng, and Fulong Wang. 2025. [Jasper and stella: distillation of sota embedding models](#).
- Yejian Zhang and Shingo Takada. 2025. [Applying LLMs to active learning: Toward cost-efficient cross-task text classification without manually labeled data](#). *International Journal of Intelligent Systems*, 2025(1):6472544.

A Comprehensive description of the descriptor schema generation pipeline

A.1 Descriptor generation pipeline

We use the 70 billion parameter instruction fine-tuned version of Llama 3.3 to label our data. We find that this model generates high-quality responses while still being practical to run on large amounts of data.

The pipeline that creates the descriptor schema is divided into three steps. First comes the *Generation* step, in which the initial descriptors are generated. This step consists of three stages. After this comes

the *Disambiguation* step. During this step, most duplicate descriptors are merged into one or split into separate descriptors. Finally comes the *Synonym merging* step during which synonymous or near-synonymous descriptors are merged.

A.2 Generation step

The generation step is divided into three stages: initial generation, rewriting, and revision. In the initial generation stage, the model is given one document and is prompted to "[d]escribe the document in aspects including, but not limited to, style, tone, genre, topic, domain, language, quality, sentiment, etc." The model is asked to generate descriptors based on what it believes is essential for understanding, summarizing, or rewriting the document. The instructions are kept purposefully vague to not guide the model too much in a predetermined direction and to capture a broad range of descriptors. For each tag, the model is also prompted to generate a definition, which should "explain its meaning and relevance to the document in 2 to 3 sentences". The purpose of the definition is to provide additional context which is needed in later stages of the pipeline.

Since our aim is to build a general annotation schema, we do not want the descriptors to be overly specific. However in practice, we noted the model had a tendency to generate overly specific descriptors. For this reason, we offer the model the option to add more detailed descriptions into a list of "specifics". These are only used in the initial generation step but are discarded later.

A rewriting stage follows the initial generation. To evaluate how well the meaning of the document is captured in the descriptors, we give the descriptors and specifics generated in the previous stage back to the model and ask it to rewrite the document that they describe.

The rewriting stage is followed by a revision stage. After generating descriptors, specifics and a rewrite based on them, the model is asked to evaluate the quality of the rewrite and revise the descriptors accordingly. The model is prompted to compare the original document and its rewrite and to generate a revised set of descriptors that would better capture the meaning of the document.

The revision stage is followed by another rewrite stage, this time using the revised descriptors as input. This is followed by another revision stage and finally one more rewrite. Although this pro-

cess could be repeated arbitrarily many times, we stop after three sets of descriptors and rewrites. To identify the set of descriptors that best captured the meaning of the original document, we embed the original document and each of its rewrites using the Stella-en-400M-v5 model [Zhang et al. \(2025\)](#), which is one of the best performing models in its size for English text clustering on the MTEB leaderboard ([Muennighoff et al., 2023](#)). The cosine similarity between each rewrite and the original is used as a measure of how much the rewrite resembles the original. The closer a rewrite is to the original document, the better the descriptors have summarized the document and thus, the better their quality.

After these steps, for each document, we only keep the descriptors that produced the best rewrite (measured by cosine similarity) and discard the rest. The specifics are also discarded at this stage.

A.3 Disambiguation step

In the disambiguation stage, we normalize all tags to lowercase and group together all definitions for a given tag. The vast majority of tags have only one definition, but some have more, with the most common ones such as "informal tone" having thousands of different definitions. The tags with more than one definition are provided to the LLM, which is prompted to either merge together synonymous or near-synonymous entries and provide one holistic descriptor or split the group into several descriptors, each with a locally unique tag and definition. To avoid overwhelming the model with extremely long inputs, large groups are split into subgroups of 20 definitions or fewer, each of which is disambiguated separately. As the model is not aware of decisions made in different instances, it can generate new, locally unique descriptors that are still duplicates globally. Therefore, this process is repeated several times until there are only a handful of duplicates left.

After 20 iterations, only four tags remain with more than one definition. The process is stopped here, since the model keeps inventing descriptors that already exists elsewhere in the data and the rate of lowering the number of duplicates approaches zero.

A.4 Synonym merging step

After disambiguation, there are still many synonymous and near-synonymous descriptors in the set because in the disambiguation step only exact tag duplicates were grouped together. For instance,

tags such as “formal tone” and “formal language” are still separate, despite being arguably synonymous in most contexts. In the synonym merging step, each descriptor is embedded with the *Stella-en-400M-v5* model and mutually nearest neighbors are provided to the LLM. The model is prompted to either keep them as separate entries or accept them as synonyms and choose one as the representative, while the other is discarded. The discarded descriptor is thus effectively merged into the other. After each iteration, new mutually nearest neighbors are identified and provided to the model. The process ends when no more merges are made.

The synonym merging step merges most of the remaining duplicate tags that survived the disambiguation step, but leaves just one tag with more than one definition. To ensure that each tag has exactly one definition, we do a forced merge, where the remaining duplicates are provided to the model, which chooses one as the representative descriptor.

B Example document and descriptors

Document

- I am now the proud owner of a beautiful Black (the fastest colour of course) GCSRT8 here in Western Australia. There is a growing number of these beasts here although I have yet to see one on the road. You seem to have a fantastic SRT8 community thing going in this forum, I only wish that there was a GCSRT8 group like you here. (If there is an active SRT8 forum/group in Oz, can someone let me have the details?) I have gathered a lot of useful info from the forum to help me with my next move regarding performance mods and have decided the first will be to change my tstat to a 180 (and maybe the fan mod - although I am not that comfortable in cutting wires etc to get it done) Can anyone please advise where I can get these thermostats in Australia? What brands are available here? Part numbers? Our local auto chain store (Supa Cheap) hardly even knows what they are let alone if they stock them! Any help would be appreciated.

Document rewrite

- G'day mates! I'm a car enthusiast from Western Australia, and I'm stoked to be part of this awesome community. I've got a GCSRT8, which I've modified with an SRT8 kit, and I'm always on the lookout for ways to improve its

performance. Recently, I picked up a 180 thermostat from Supa Cheap, and I'm thinking of doing a fan mod to keep things cool under the bonnet. I've heard it's a bit of a tricky job, but I'm keen to give it a go. Has anyone else out there done this mod? I'd love to get some advice from you blokes with some technical expertise. I've been doing some research, and I've got the part numbers written down, but I'm not sure about cutting wires and whatnot. I've been to the local auto chain store, and they didn't seem to know much about performance modification, so I thought I'd come to the car community for some help. By the way, what's the fastest colour for a GCSRT8? Just kidding, mate! But seriously, thanks in advance for any tips or advice you can offer. I'm really appreciative of all the knowledge and enthusiasm on this forum, and I'm excited to hear from you all.

Raw descriptors

- car enthusiast; The document is written by a car enthusiast, indicating a strong interest in vehicles and their modification. The author's passion for cars is evident in their language and tone.
- technical; The document discusses technical aspects of car modification, such as thermostats and fan mods. This indicates a focus on the mechanical details of the vehicle.
- Australian; The document is set in Western Australia, and the author is seeking information specific to their location. This context is important for understanding the author's needs and concerns.
- informal; The document has a casual and friendly tone, indicating that the author is approachable and enthusiastic about the topic. The use of colloquial language and expressions conveys a sense of camaraderie.
- modification-focused; The document discusses plans for modifying the vehicle's performance, including changing the thermostat and potentially installing a fan mod. This indicates a focus on improving the vehicle's capabilities.

- seeking advice; The author is seeking guidance and recommendations from the community, demonstrating a willingness to learn and take advice from others. This shows that the author values the opinions and expertise of others.
- personal; The author is a proud owner of a GC-SRT8 and is seeking to modify and improve their vehicle, indicating a personal investment in the topic. The author's ownership and enthusiasm are central to the document.
- regional; The author is seeking information specific to Australia, including availability of parts and local resources. This context is crucial for understanding the author's needs and concerns.
- appreciative; The author expresses gratitude for the community's potential help and advice, demonstrating a positive and respectful attitude. This tone is important for building relationships and trust within the community.
- enthusiastic; The author is excited about their vehicle and the potential modifications, conveying a sense of eagerness and anticipation.
- community-oriented; The author is seeking to connect with others who share their interests and is looking for a sense of community and belonging.

Harmonized descriptors

- automotive enthusiast; The author is an automotive enthusiast with a strong interest in modifying and improving vehicles, possessing deep knowledge of cars and trucks, and valuing high-quality service.
- automotive performance; The document focuses on the automotive domain, discussing vehicle performance, modifications, and technical aspects.
- western australia context; The text is specific to Western Australia, highlighting the regional focus of the document.
- informal social; The tone of the document is casual and friendly, indicating a personal or social relationship between the writer and the reader, and describes a situation that occurs

in a social setting, making it interesting and engaging, and involves colleagues in a light-hearted activity.

- performance mods; The document discusses various modifications made to vehicles to improve their performance, including upgrades to engines, transmissions, suspensions, and other parts, as well as customization and DIY efforts.
- user seeking advice; This descriptor captures the purpose of the post, which is for the user to seek information or advice from others.
- passion and identity; The author's passion for cars is a central aspect of their identity and is reflected in their language and experiences, with a focus on modifying and customizing their vehicle.
- relevance to australia; The document is specific to the Australian context, making it relevant to an Australian audience, with references to local policies, laws, and regulations.
- polite and appreciative tone; The tone is polite and appreciative, with the authors expressing gratitude for any help they might receive.
- enthusiastic mood; The tone is positive and enthusiastic, conveying excitement and anticipation.
- online community seeker; The author is looking for a community where they can express themselves freely and engage in discussions with others who share similar interests. This descriptor emphasizes the author's desire for connection and dialogue.

Specifics

- GCSRT8
- SRT8
- Western Australia
- Supa Cheap
- 180 thermostat
- fan mod
- part numbers

- performance modification
- car community
- technical expertise
- fastest colour
- cutting wires
- forum
- local auto chain store

C Query phrases

The exact phrases used to query documents. The phrasing of the topic and format queries are derived from the WebOrganizer paper.

• Formats

- About (Org.); The document is formatted as an organizational About page, typically containing a self-description or introduction by an organization such as a company, university, government agency, or non-profit, but the content is not necessarily verified and may contain self-promotion
- About (Personal); The document is formatted as a personal About page, typically containing a self-description, introduction, or profile information on a personal or hobby website
- Academic Writing; The document is formatted as academic writing, such as a research paper, paper abstract, thesis, or literature review
- Audio Transcript; The document is formatted as an audio transcript, providing a written record of spoken language, such as an interview, court hearing, movie, podcast, lecture, or speech
- Comment Section; The document is formatted as a comment section or discussion forum with multiple posts or comments, such as a Reddit thread or a comment section on a news article or blog
- Content Listing; The document is formatted as a content listing, providing an overview of content for navigation, such as a sitemap, product catalog, search results page, or news listing with short article snippets
- Creative Writing; The document is formatted as creative writing, such as a short story, novel chapter, poem, or song lyrics
- Documentation; The document is formatted as documentation, such as technical writing, API documentation, README files, or source code, intended for developers or experts rather than end-users
- FAQ; The document is formatted as a Frequently Asked Questions page
- Knowledge Article; The document is formatted as a knowledge article, written in an objective and neutral style and published on a moderated platform or by a reputable source
- Legal Notices; The document is formatted as legal notices, such as terms of service, legal disclaimers, privacy policies, or license agreements
- Listicle; The document is formatted as a listicle, presenting article or blog content in the form of a list, such as a “Top 10” list or “4 best places to visit in X”, but excluding navigational lists of site content
- News (Org.); The document is formatted as organizational news or announcements, such as a press release or blog post by a company, university, government agency, or non-profit organization
- News Article; The document is formatted as a news article, written by journalists about current events and published by a news organization, but excluding long reads, profiles, editorials, journalistic essays, and newspaper interviews
- Nonfiction Writing; The document is formatted as nonfiction writing, such as a long read, profile, editorial, essay, obituary, memoir, or other professional non-fiction text
- Personal Blog; The document is formatted as a personal blog, typically written by an individual and relating personal experiences or opinions
- Product Page; The document is formatted as a product page, typically containing descriptions or promotions for a product or service, including products in a wider sense such as university course descriptions

- Q&A Forum; The document is formatted as a Q&A forum, with an explicit question-and-answer structure, such as Quora or Stack Exchange
- Spam / Ads; The document is formatted as spam or advertising, consisting primarily of spam content, SEO keyword stuffing, short online ads, or content with no apparent purpose
- Structured Data; The document is formatted as structured data, containing multiple data entries with a common structure, such as a table, datasheet, movie database, glossary, dictionary, JSON file, CSV file, or XML file
- Customer Support; The document is formatted as customer support content, produced by an organization for a general audience, such as a troubleshooting guide
- Truncated; The document is formatted as truncated or incomplete content, such as a paywalled page, login-gated page, incomplete page, video description, or image gallery, but excluding pages with multiple snippets of truncated articles
- Tutorial; The document is formatted as a tutorial or how-to guide, such as a cooking recipe, DIY instructions, WikiHow page, or Khan Academy course, and must contain the actual tutorial content
- User Review; The document is formatted as a user review, such as a review posted by users on Yelp, TripAdvisor, or similar platforms
- Entertainment; The document's topic is entertainment, including music, movies, TV shows, videos, celebrities, humor, nightlife but excluding music or film if discussed as art
- Fashion & Beauty; The document's topic is fashion and beauty, including clothing, accessories, and cosmetics
- Finance & Business; The document's topic is finance and business, including taxes, regulations, investments, insurance, credit cards, personal finance, corporate communication, marketing, and human resources
- Food & Dining; The document's topic is food and dining, including recipes, groceries, beverages, and restaurants, but excluding nutritional sciences
- Games; The document's topic is games, including video games, board games, and gambling
- Hardware; The document's topic is hardware, including computer hardware, phones, televisions, and other consumer electronics
- Health; The document's topic is health, including medicine, wellness, mental health, veterinary science, and nutritional science, but excluding health insurance
- History; The document's topic is history, including geography and archaeology
- Home & Hobbies; The document's topic is home and hobbies, including real estate, renting, relocation, furniture, appliances, home improvement, DIY, gardening, pets, toys, and collecting
- Industrial; The document's topic is industrial activity, including mining, agriculture, manufacturing, utilities, construction, raw materials, industrial goods, chemicals, and textiles, but excluding general business topics and business finance
- Literature; The document's topic is literature, including literary criticism, linguistics, philosophy, and related subjects in the humanities, but excluding texts whose literary style is incidental to their topic

• Topics

- Adult; The document's topic is adult content, including porn and other explicit content
- Art & Design; The document's topic is art and design, including architecture
- Crime & Law; The document's topic is crime and law, including law enforcement (excluding financial crime, litigation, social issues and the legislative process)
- Education & Jobs; The document's topic is education and jobs, including pedagogy, training & certification, and academia

- Politics; The document's topic is politics, including social issues, political campaigns, the legislative process, geopolitics, protests, and activism
- Religion; The document's topic is religion, including spirituality
- Science & Technology; The document's topic is science and technology, including physics, chemistry, biology, environmental science, mathematics, statistics, biotech, and engineering
- Social Life; The document's topic is social life, including family, friends, relationships, and community, but excluding specific social activities that fall under other topics
- Software; The document's topic is software, including the use of software and the internet
- Software Development; The document's topic is software development, including algorithms, coding, and web development
- Sports & Fitness; The document's topic is sports and fitness, including martial arts, motor sports, outdoor activities, and sports equipment
- Transportation; The document's topic is transportation, including cars and other vehicles, taxis, public transportation, traffic, commuting, aviation, rail, shipping, and logistics
- Travel; The document's topic is travel, including hospitality, hotels, sight-seeing, and cruises, but excluding detailed descriptions of tourist destinations

- **Sarcasm**

- sarcasm; the document contains sarcastic humor and irony

D Prompts

This Appendix contains all the prompts used in the pipeline.

D.1 Initial prompt

System prompt:

```
##Instruction:
You will be given a document. Your task is to create a comprehensive
list of descriptors - words or phrases that distill the meaning,
tone, style, genre, topics, and other characteristics of the
document. Do not focus solely on the topics; also include phrases
that describe the tone and style.
The descriptors you generate should be useful for two purposes:
1. Rewriting the document in a way that is faithful to the original.
2. Classifying the document in a large variety of NLP classification
tasks.
##Requirements:
1. Generate Descriptors:
- Describe the document in aspects including, but not limited to,
style, tone, genre, topic, domain, language, quality, sentiment
etc. Describe anything that you believe is essential for
understanding, summarizing, or rewriting the document.
- If necessary, you can add minute details into the "specifics" list.
This list can contain individual words, phrases and other details
that do not necessarily describe the document but are important
for rewriting it.
2. Descriptor Details:
- Descriptors can be single words or multi-word phrases. They should
typically not be longer than 3 words.
- Each descriptor should describe only one aspect of the document.
Multi-aspect descriptors should be split into separate descriptors
.
- The list should be so extensive that, given only the list, a
competent person should be able to rewrite the original document.
- Do not repeat descriptors and avoid near-duplicates.
3. Order:
- Order the descriptors in order of importance: the most important
descriptors should be listed first.
- Importance is determined by how essential the descriptor is for
reconstructing, understanding, or summarizing the document.
4. Descriptor Explanations:
- For each descriptor, explain its meaning and relevance to the
document in 2 to 3 sentences.
- The items in the specifics list do not need to be explained.
5. Format:
- The output must be formatted as a JSON object. Follow this
formatting exactly.
- Never add any preamble or anything else outside the JSON object.
- Separate the descriptors and their explanations with a semicolon
';'. Do NOT include a semicolon anywhere else in the descriptor.
- Example: {"descriptors": ["<descriptor>; <explanation>", "<
descriptor>; <explanation>", "<descriptor>; <explanation>", ...],
```

```
"specifics": ["<specifics>", "<specifics>", "<specifics>", ...]}
```

User prompt:

```
{document}
```

D.2 Rewrite prompt

System prompt:

```
##Instruction:
You will be given a list of descriptors and specifics:
- Descriptors: Describe the document in aspects including, but not
  limited to, style, tone, genre, topic, domain, language, quality,
  sentiment, etc.
- Each descriptor is followed by an explanation of its meaning and
  relevance to the document.
- The descriptors are ordered in order of importance: the most
  important descriptors are listed first.
- Specifics: Minute details such as individual words and phrases,
  emphasis, structure, etc. to help in rewriting the document.
##Your Task:
Reconstruct the original document as accurately as possible using
  both the descriptors and specifics provided. Do not describe the
  document; instead, produce the document itself.
##Requirements:
1. Utilize All Descriptors:
- Incorporate all descriptors and specifics into your writing.
- Ensure that the content, tone, style, and structure align with the
  descriptors.
- The descriptors are ordered in order of importance: the most
  important descriptors are listed first.
2. Format:
- The output must be formatted as a JSON object. Follow this
  formatting exactly.
- Never add any preamble or anything else outside the JSON object.
- Example: {"text": "<rewritten version of the document>"}
```

User prompt:

```
*Descriptors*:
<start>{descriptors}<end>
*Specifics*:
<start>{specifics}<end>
```

D.3 Revision prompt

System prompt:

```
##Instruction:
You will be given:
1. Two Documents:
- Original Document: The initial text.
- Rewritten Document: An attempt to recreate the original document
  based solely on provided descriptors and specifics.
```

2. A List of Descriptors and Specifics (describing the original document):
 - Descriptors: Describe the document in aspects including, but not limited to, style, tone, genre, topic, domain, language, quality, sentiment, etc.
 - Each descriptor is followed by an explanation of its meaning and relevance to the document.
 - The descriptors are ordered in order of importance: the most important descriptors are listed first.
 - Specifics: Minute details such as individual words and phrases, emphasis, structure, etc. to help in rewriting the document.
- ##Your Task:
1. Critique the Rewritten Document:
 - Analyze the rewritten document.
 - Point out ways in which the rewrite differs from the original. Point out specific sentences or phrases that diverge meaningfully from the original, if applicable.
 - Focus on discrepancies in content, tone, style, structure, and any missing or misrepresented elements.
 2. Revise the Descriptors and Specifics:
 - Adjust and revise the descriptors and specifics to help in creating a more faithful rewrite of the original document and for classifying it in a large variety of NLP classification tasks.
 - Each descriptor should describe only one aspect of the document. Multi-aspect descriptors should be split into separate descriptors.
 - The specifics list can contain individual words, phrases and other details that do not necessarily describe the document but are important for rewriting it.
 - The revised lists should not contain any direct or indirect references or comparisons to the original lists. They should stand on their own.
 - For each descriptor, explain its meaning and relevance to the document in 2 to 3 sentences. The specifics do not need to be explained.
- ##Requirements:
1. Specificity: Provide detailed observations in your critique.
 2. Clarity: Ensure the revised descriptors are clear and precise.
 3. Completeness: Cover all aspects where the rewritten document deviates from the original.
 4. Conciseness: Do not make the descriptors overly long. They should typically not be longer than 3 words. Do not repeat descriptors and avoid near-duplicates.
 5. Order: Order the descriptors in order of importance: the most important descriptors should be listed first.
 6. Format:
 - The output must be formatted as a JSON object. Follow this formatting exactly.
 - Never add any preamble or anything else outside the JSON object.
 - Separate the descriptors and their explanations with a semicolon ';'. Do NOT include a semicolon anywhere else in the descriptor.

```
- Example: {"differences": "<differences between original and rewrite>",
"descriptors": ["<descriptor>; <explanation>", "<descriptor>; <explanation>", ...], "specifics":
["<specifics>", "<specifics>", "<specifics>", ...]}
```

User prompt:

```
{
*Original Document*:
<start>{original_document}<end>
*Rewritten Document*:
<start>{rewritten_document}<end>
*Descriptors*:
<start>{descriptors}<end>
*Specifics*:
<start>{specifics}<end>
}
```

D.4 Disambiguation prompt

System prompt:

You will be given a descriptor and its list of explainers. The descriptor describes a document and the explainers provide additional context to the descriptor.

Each explainer line is prefixed with its INDEX in square brackets, like: [0] text...

Your job is to merge/split explainers into semantically consistent groups *and return the INDICES of the original explainers you placed into each group*. Do not invent new indices, and use each index exactly once across all groups.

\#\# Instructions

- 1) Read the descriptor and explainers (each line is like: [<index>] <explainer text>).
- 2) Compare the explainers to each other and consider:
 - Do they point to the same kind of interpretation, rationale, or feature?
 - Do they reflect a consistent understanding or application of the descriptor?
- 3) Based on this comparison, create one or more semantically coherent groups.
- 4) For each group, provide:
 - group_descriptor (<= 3 words where possible. Coherent with the explainers in this group.)
 - group_explainer (Concise, canonical summary that covers the meaning of all explainers in this group)
 - original_explainer_indices_in_this_group: an array of the *indices* you grouped here (e.g., [0, 5, 7]).

Notes:

- Do NOT drop any original explainer indices. Every original explainer index must appear exactly once across groups.

- You may keep the original descriptor for one group; newly invented descriptors are allowed for other groups.
- Make minimal changes to the original explainers when forming group_explainers; focus on summarizing and unifying the meanings.
- Descriptors and explainers must be self-contained -- they must not contain references or comparisons to the original descriptors or explainers or the other groups.

\#\# Output format

Return a single valid JSON object with this exact structure (no extra text or commentary of any kind):

```
{
  "original_descriptor": "<original descriptor>",
  "groups": [
    {
      "group_descriptor": "<group descriptor>",
      "group_explainer": "<group explainer>",
      "original_explainer_indices_in_this_group":
        [<index_a>, <index_b>, ...]
    }
  ]
}
```

User prompt:

```
Descriptor: {descriptor}
Explainers (each prefixed by its INDEX):
{explainers}
```

D.5 Synonym merging prompt

System prompt:

You are deciding whether two descriptor-explainer pairs, A and B, refer to the same concept.

The descriptors describe a document, while the explainer provides additional context.

Treat them as synonyms (set is_synonym=true) if any of the following is true:

- They refer to the same concept or entity despite different wording.
- They deviate slightly in specificity but still cover the same core meaning.
- They differ in phrasing but could be used interchangeably in most contexts.
- They differ in word order, punctuation or spelling but mean the same thing.
- They list similar things in different order.

Do NOT treat them as synonyms (set is_synonym=false) if any of the following is true:

- They refer to different concepts/entities or mutually exclusive categories.
- One is substantially broader/narrower in scope or specificity.
- The explainer introduces a difference in meaning even when descriptors are similar.

In edge cases, prefer to NOT treat as synonyms.

If synonyms:

- choose which to keep: "A" or "B" (prefer clearer, more general phrasing);
- set keep to the chosen label and drop to the other.

If not synonyms:

- Still return keep and drop as A and B (order doesn't matter).

Return ONLY JSON with keys: is_synonym (bool), keep ("A"|"B"), drop ("A"|"B").

Do not rewrite or invent descriptors.

User prompt:

```
A: descriptor: {descriptorA} explainer: {explainerA}
=====
B: descriptor: {descriptorB} explainer: {explainerB}
```

D.6 Force merging prompt

System prompt:

You are selecting a representative pair among candidates that share the same descriptor.

The descriptor describes a document, while the explainer provides additional context.

Your task is to choose the most representative pair among the given pairs.

Return the integer index of the pair to keep in the field 'keep'

When choosing the representative pair, prefer concise, general, commonly used phrasing that covers the meaning of all pairs.

Return ONLY JSON with: keep (int). Do not invent or rewrite anything.

User prompt:

```
index: {i} | descriptor: {descriptorA} | explainer: {explainerA}
index: {j} | descriptor: {descriptorA} | explainer: {explainerB}
...
```

D.7 Harmonization prompt

System prompt:

You are a careful ontology assistant.

Task: Given a query pair in the form 'descriptor; explainer' and a list of schema candidates, pick the single candidate that is synonymous or near-synonymous in most contexts to the query.

If many candidates are synonymous, choose the one that is the closest match.

If you find a suitable candidate, respond with is_synonym=true and set chosen_index to exactly the chosen candidate's index.

If none are acceptable, return is_synonym=false.

Output MUST be JSON only.

User prompt:

```

Query: {descriptor}; {explainer}
Candidates:
{i} {candidateA}
{j} {candidateB}
...

```

E Annotation results

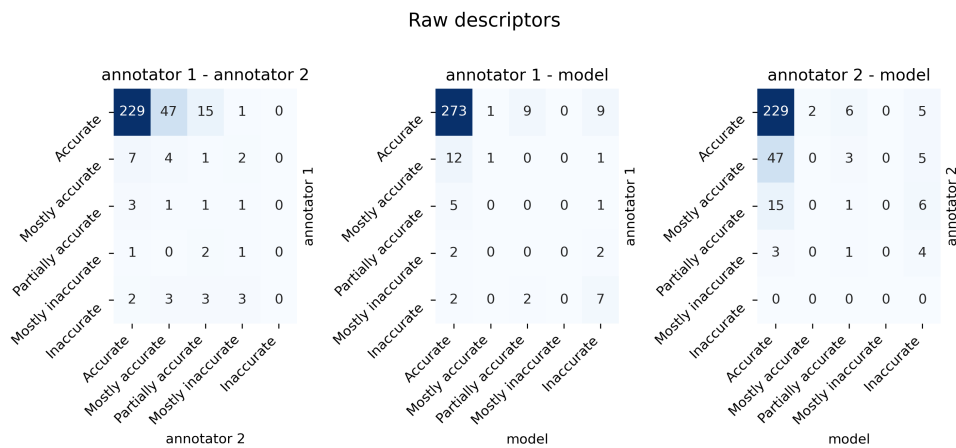


Figure 6: Confusion matrices for descriptor accuracy annotations with raw descriptors.

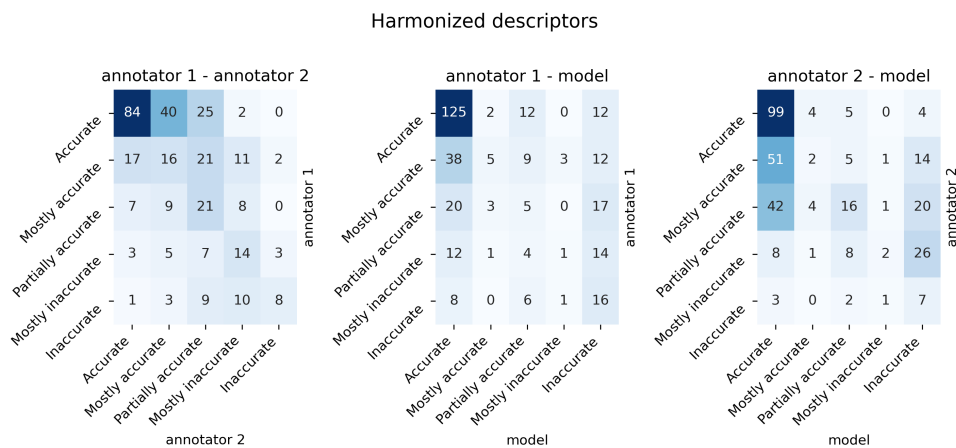


Figure 7: Confusion matrices for descriptor accuracy annotations with harmonized descriptors.

F Descriptor aspect categories

Main category	Examples
Content & Subject Matter	e.g. topic, domain, subtopic, intent, cultural context, temporal relevance
Structure & Format	e.g. genre, document type (blog, news, etc.), media format, length
Style & Tone	e.g. formality, tone, technicality (e.g. jargon), voice (active/passive), syntax
Quality & Trustworthiness	e.g. factuality, authority, originality, spam, bias, depth (superficial/in-depth)
Audience & Engagement	e.g. target audience, engagement level, monetization, call-to-action, shareability
Linguistic Features	e.g. readability, sentiment, figurative language, multilinguality, repetition
Ethical & Legal	e.g. toxicity, misinformation, legal compliance, accessibility, dark patterns
Other	miscellaneous, not fitting into other categories

Table 5: Broad categories in which we categorize descriptors with examples of their contents

The descriptor categories were created through an iterative process involving discussion between the authors and with the help of an LLM. The categories reflect the type of descriptors that we observed to exist in the data and the kinds of descriptors that we would hope to exist and that researchers in the field of corpus linguistics and NLP might find relevant.

G Classification results

Task	Features	Macro-F1 [95% CI]	W-F1	$ V $
FineWeb Edu	Descriptors (Harmonized)	0.70 [0.69, 0.71]	0.92	278,093
FineWeb Edu	Descriptors (Raw)	0.76 [0.75, 0.77]	0.93	351,003
FineWeb Edu	TF-IDF	0.75 [0.74, 0.77]	0.94	50,000
Topic	Descriptors (Harmonized)	0.63 [0.62, 0.64]	0.64	278,093
Topic	Descriptors (Raw)	0.65 [0.64, 0.65]	0.66	351,003
Topic	TF-IDF	0.73 [0.72, 0.74]	0.75	50,000
Format	Descriptors (Harmonized)	0.44 [0.43, 0.45]	0.53	278,093
Format	Descriptors (Raw)	0.48 [0.47, 0.49]	0.57	351,003
Format	TF-IDF	0.49 [0.48, 0.50]	0.61	50,000

Table 6: Classification F1 comparing harmonized vs. raw descriptors, with TF-IDF as a text-based reference. Confidence intervals from 1000 bootstrap resamples of the test set. Descriptor configs use minimum frequency 1 (full vocabulary); TF-IDF capped at 50,000 features. $|V|$ = vocabulary size.

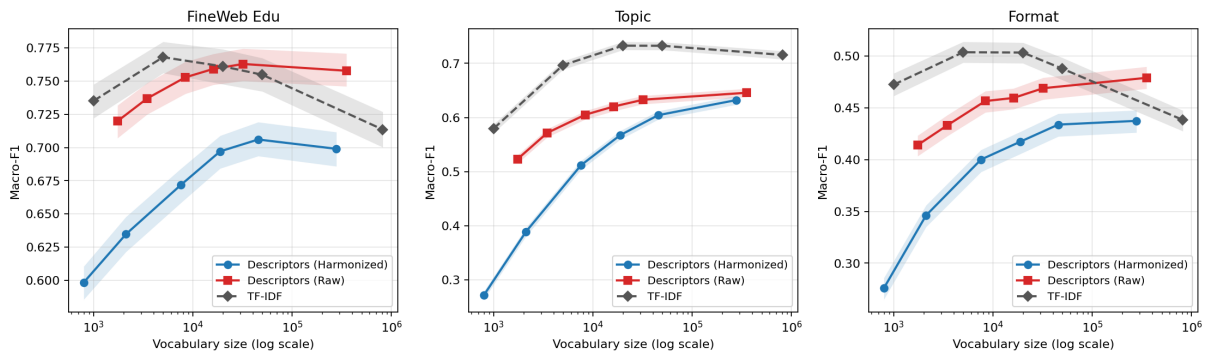


Figure 8: Macro-F1 as a function of feature-set size for harmonized descriptors, raw descriptors, and TF-IDF, across the three tasks. Descriptor vocabulary size is swept by minimum token frequency; TF-IDF by maximum vocabulary size. Shaded bands are 95% bootstrap confidence intervals.

Class	Desc. (Harm.)			Desc. (Raw)			TF-IDF			N
	Scan%	Lift	Prec.	Scan%	Lift	Prec.	Scan%	Lift	Prec.	
edu	16.9%	4.7	37.9%	14.5%	5.5	44.0%	10.9%	7.4	58.7%	1,597
not-edu	74.4%	1.1	98.9%	74.3%	1.1	99.1%	73.8%	1.1	99.7%	18,403
Weighted mean	69.8%	1.4	94.0%	69.5%	1.4	94.7%	68.8%	1.6	96.5%	20,000

Table 7: Per-class retrieval at 80% recall, FineWeb Edu. Scan% = fraction of test set scanned to reach the target recall (lower better); Lift = precision/base rate; Prec. = precision at that point; N = support. Min. frequency 1; TF-IDF capped at 50,000. Sorted by harmonized Scan%.

Class	Desc. (Harm.)			Desc. (Raw)			TF-IDF			N
	Scan%	Lift	Prec.	Scan%	Lift	Prec.	Scan%	Lift	Prec.	
edu	2.0%	9.8	77.9%	2.0%	10.2	81.8%	1.7%	11.9	94.7%	1,597
not-edu	18.4%	1.1	99.9%	18.4%	1.1	100.0%	18.4%	1.1	100.0%	18,403
Weighted mean	17.1%	1.8	98.1%	17.1%	1.8	98.5%	17.1%	1.9	99.5%	20,000

Table 8: Per-class retrieval at 20% recall, FineWeb Edu. Columns as in Table 7.

Class	Desc. (Harm.)			Desc. (Raw)			TF-IDF			N
	Scan%	Lift	Prec.	Scan%	Lift	Prec.	Scan%	Lift	Prec.	
Adult	2.2%	35.7	26.1%	5.6%	14.2	10.4%	2.5%	32.3	23.6%	146
Religion	4.2%	18.9	39.2%	3.8%	21.3	44.0%	3.0%	26.3	54.4%	414
Travel	4.5%	17.8	42.3%	4.3%	18.5	43.9%	3.0%	26.7	63.2%	474
Software Dev.	4.5%	17.6	29.6%	4.9%	16.5	27.7%	2.9%	27.4	46.1%	336
History	7.4%	10.8	13.0%	4.6%	17.5	21.1%	3.4%	23.4	28.2%	241
Crime & Law	8.0%	10.0	35.9%	7.7%	10.4	37.2%	4.9%	16.4	59.1%	719
Hardware	8.3%	9.7	24.6%	8.1%	10.0	25.3%	3.9%	20.5	52.0%	508
Games	8.6%	9.3	28.2%	10.0%	8.0	24.3%	5.8%	13.9	42.4%	608
Fashion & Beauty	8.7%	9.2	34.9%	9.2%	8.7	33.0%	4.2%	18.9	71.9%	760
Science & Tech.	8.8%	9.1	36.9%	9.0%	8.9	36.1%	6.7%	12.0	48.4%	808
Food & Dining	9.2%	8.7	38.7%	10.0%	8.0	35.8%	4.5%	18.0	80.3%	894
Health	9.7%	8.3	50.7%	8.5%	9.4	57.4%	6.7%	11.9	73.0%	1,225
Software	11.0%	7.3	21.0%	9.5%	8.4	24.3%	5.8%	13.8	39.9%	579
Sports & Fitness	11.2%	7.1	56.8%	10.6%	7.5	59.9%	7.7%	10.4	83.1%	1,595
Politics	11.3%	7.1	46.2%	11.1%	7.2	47.1%	8.5%	9.4	61.3%	1,305
Entertainment	11.8%	6.8	50.6%	14.0%	5.7	42.7%	8.4%	9.5	70.9%	1,497
Education & Jobs	12.8%	6.3	37.0%	12.9%	6.2	36.7%	7.6%	10.6	62.3%	1,180
Industrial	13.3%	6.0	17.5%	11.3%	7.1	20.6%	6.1%	13.1	37.7%	578
Literature	13.7%	5.8	22.4%	14.2%	5.6	21.6%	8.3%	9.6	36.8%	766
Art & Design	14.4%	5.5	17.2%	10.5%	7.6	23.6%	5.0%	15.9	49.4%	620
Finance & Business	16.0%	5.0	37.5%	14.7%	5.5	41.0%	10.4%	7.7	57.7%	1,503
Home & Hobbies	16.7%	4.8	32.9%	17.4%	4.6	31.6%	10.0%	8.0	54.9%	1,374
Transportation	16.8%	4.8	16.0%	18.2%	4.4	14.8%	5.0%	15.8	53.2%	672
Social Life	18.7%	4.3	25.7%	17.0%	4.7	28.2%	12.7%	6.3	37.8%	1,198
Weighted mean	11.8%	7.8	36.3%	11.6%	7.9	36.8%	7.2%	12.9	58.4%	20,000

Table 9: Per-class retrieval at 80% recall, Topic. Columns as in Table 7.

Class	Desc. (Harm.)			Desc. (Raw)			TF-IDF			N
	Scan%	Lift	Prec.	Scan%	Lift	Prec.	Scan%	Lift	Prec.	
Adult	0.1%	132.6	96.8%	0.2%	128.4	93.8%	0.2%	128.4	93.8%	146
History	0.3%	63.5	76.6%	0.3%	58.1	70.0%	0.3%	65.6	79.0%	241
Software Dev.	0.5%	44.0	73.9%	0.5%	40.9	68.7%	0.4%	46.0	77.3%	336
Religion	0.5%	43.1	89.2%	0.5%	39.7	82.2%	0.5%	42.7	88.3%	414
Travel	0.6%	36.8	87.2%	0.6%	33.4	79.2%	0.5%	39.3	93.1%	474
Hardware	0.6%	33.7	85.7%	0.7%	28.9	73.4%	0.6%	32.6	82.9%	508
Games	0.6%	31.4	95.3%	0.7%	29.5	89.7%	0.6%	32.4	98.4%	608
Art & Design	0.8%	25.8	80.0%	0.9%	22.9	70.9%	0.7%	28.6	88.6%	620
Crime & Law	0.8%	25.2	90.6%	0.8%	24.3	87.3%	0.8%	26.5	95.4%	719
Fashion & Beauty	0.8%	24.1	91.6%	0.8%	23.5	89.4%	0.8%	26.0	98.7%	760
Transportation	0.9%	22.8	76.7%	1.0%	20.0	67.2%	0.8%	26.1	87.7%	672
Food & Dining	0.9%	21.8	97.3%	0.9%	21.8	97.3%	0.9%	21.9	97.8%	894
Science & Tech.	0.9%	21.6	87.1%	1.1%	18.7	75.3%	0.9%	21.4	86.6%	808
Software	1.0%	20.9	60.4%	1.0%	19.8	57.4%	0.8%	24.6	71.2%	579
Industrial	1.0%	20.6	59.5%	1.0%	19.5	56.3%	0.7%	26.9	77.8%	578
Literature	1.3%	15.3	58.8%	1.3%	15.5	59.5%	0.9%	22.8	87.5%	766
Health	1.4%	14.7	90.1%	1.4%	14.7	90.1%	1.3%	15.3	93.5%	1,225
Education & Jobs	1.4%	14.5	85.8%	1.4%	14.1	83.1%	1.3%	15.5	91.5%	1,180
Politics	1.6%	12.7	82.9%	1.6%	12.1	78.8%	1.4%	14.2	92.9%	1,305
Sports & Fitness	1.6%	12.5	100.0%	1.6%	12.3	98.5%	1.6%	12.4	99.1%	1,595
Entertainment	1.6%	12.3	92.3%	1.7%	12.0	89.8%	1.6%	12.9	96.8%	1,497
Home & Hobbies	1.7%	11.8	81.4%	1.8%	11.3	77.7%	1.5%	13.6	93.5%	1,374
Finance & Business	1.7%	11.6	87.0%	1.9%	10.9	81.6%	1.6%	12.4	93.5%	1,503
Social Life	2.0%	9.8	58.8%	1.9%	10.4	62.5%	1.4%	13.9	83.0%	1,198
Weighted mean	1.3%	19.8	83.8%	1.3%	18.8	80.1%	1.1%	21.5	91.2%	20,000

Table 10: Per-class retrieval at 20% recall, Topic. Columns as in Table 7.

Class	Desc. (Harm.)			Desc. (Raw)			TF-IDF			N
	Scan%	Lift	Prec.	Scan%	Lift	Prec.	Scan%	Lift	Prec.	
Legal Notices	1.5%	55.4	34.4%	1.2%	69.2	42.9%	1.2%	65.0	40.3%	124
Documentation	4.6%	17.5	8.9%	3.5%	22.8	11.6%	1.4%	58.0	29.6%	102
Academic Writing	5.0%	16.2	25.9%	3.5%	22.6	36.1%	2.9%	27.6	44.1%	319
Creative Writing	5.9%	13.6	16.4%	5.3%	15.2	18.4%	7.1%	11.2	13.6%	242
User Review	9.8%	8.1	15.2%	7.9%	10.1	18.8%	7.7%	10.4	19.3%	372
Comment Section	12.3%	6.5	24.5%	12.1%	6.6	24.9%	8.1%	9.9	37.0%	751
About (Pers.)	13.0%	6.2	11.4%	10.1%	7.9	14.6%	5.3%	15.2	28.1%	369
Tutorial	14.6%	5.5	25.2%	11.0%	7.3	33.4%	8.3%	9.6	44.3%	920
Customer Support	15.7%	5.1	8.1%	12.8%	6.3	9.9%	7.2%	11.2	17.8%	318
About (Org.)	16.1%	5.0	17.5%	14.9%	5.4	18.9%	9.5%	8.4	29.7%	704
Spam / Ads	20.5%	3.9	9.3%	14.9%	5.4	12.7%	4.9%	16.4	38.8%	473
Q&A Forum	22.5%	3.6	2.7%	14.0%	5.8	4.4%	6.0%	13.4	10.2%	153
News Article	23.2%	3.5	53.3%	22.0%	3.6	56.1%	16.7%	4.8	74.3%	3,092
News (Org.)	23.7%	3.4	20.8%	20.6%	3.9	23.8%	16.5%	4.9	29.8%	1,227
Personal Blog	23.7%	3.4	50.2%	25.9%	3.1	45.9%	24.0%	3.3	49.5%	2,974
Knowledge Article	24.4%	3.3	14.4%	20.5%	3.9	17.1%	11.8%	6.8	29.6%	877
Product Page	25.1%	3.2	54.9%	26.1%	3.1	52.9%	21.4%	3.7	64.4%	3,444
Structured Data	25.3%	3.2	5.1%	24.6%	3.2	5.2%	9.5%	8.4	13.4%	320
Audio Transcript	28.1%	2.8	4.3%	23.6%	3.4	5.1%	13.4%	6.0	8.9%	300
Nonfiction Writing	31.0%	2.6	14.9%	29.5%	2.7	15.6%	19.7%	4.1	23.4%	1,152
Listicle	34.7%	2.3	5.7%	25.9%	3.1	7.6%	19.0%	4.2	10.4%	492
Truncated	47.6%	1.7	3.1%	41.7%	1.9	3.6%	13.5%	5.9	11.0%	371
FAQ	49.2%	1.6	0.7%	29.9%	2.7	1.1%	13.4%	6.1	2.5%	84
Content Listing	52.5%	1.5	6.2%	44.4%	1.8	7.4%	24.4%	3.3	13.4%	820
Weighted mean	23.9%	4.3	33.0%	22.3%	4.9	33.8%	16.3%	6.8	43.7%	20,000

Table 11: Per-class retrieval at 80% recall, Format. Columns as in Table 7.

Class	Desc. (Harm.)			Desc. (Raw)			TF-IDF			N
	Scan%	Lift	Prec.	Scan%	Lift	Prec.	Scan%	Lift	Prec.	
Legal Notices	0.1%	161.3	100.0%	0.1%	155.1	96.2%	0.1%	155.1	96.2%	124
Documentation	0.3%	79.2	40.4%	0.1%	152.5	77.8%	0.2%	117.6	60.0%	102
FAQ	0.3%	68.6	28.8%	1.2%	17.5	7.4%	0.2%	88.0	37.0%	84
Audio Transcript	0.4%	51.9	77.9%	0.6%	34.5	51.7%	0.4%	53.3	80.0%	300
Creative Writing	0.4%	47.6	57.6%	0.5%	43.5	52.7%	0.3%	75.0	90.7%	242
Academic Writing	0.5%	45.1	71.9%	0.4%	50.8	81.0%	0.4%	51.4	82.0%	319
Q&A Forum	0.6%	36.5	27.9%	0.7%	27.2	20.8%	0.3%	63.3	48.4%	153
User Review	0.6%	31.8	59.1%	0.7%	28.2	52.4%	0.6%	32.5	60.5%	372
About (Pers.)	0.7%	31.1	57.4%	0.6%	33.1	61.2%	0.5%	37.8	69.8%	369
Spam / Ads	0.7%	29.3	69.3%	0.8%	24.8	58.6%	0.5%	37.9	89.6%	473
Customer Support	0.8%	24.8	39.5%	0.8%	25.0	39.8%	0.5%	38.3	61.0%	318
Structured Data	0.8%	24.4	39.0%	0.8%	23.4	37.4%	0.4%	48.2	77.1%	320
Comment Section	1.0%	19.8	74.4%	1.0%	19.9	74.8%	0.8%	23.7	88.8%	751
Tutorial	1.1%	18.0	82.9%	1.1%	17.8	81.8%	1.1%	18.3	84.4%	920
About (Org.)	1.4%	14.8	52.0%	1.4%	14.3	50.4%	1.0%	19.4	68.5%	704
Knowledge Article	1.6%	12.4	54.3%	1.7%	11.9	52.4%	1.2%	16.4	72.1%	877
News (Org.)	1.8%	10.9	66.8%	2.0%	9.8	60.0%	1.7%	11.9	73.0%	1,227
Listicle	2.0%	10.0	24.5%	2.1%	9.6	23.7%	1.2%	16.2	39.8%	492
Content Listing	2.7%	7.5	30.8%	3.4%	5.8	23.8%	1.8%	11.1	45.4%	820
Truncated	3.0%	6.8	12.7%	2.5%	8.0	14.8%	0.5%	37.4	69.4%	371
Nonfiction Writing	3.1%	6.6	37.8%	2.9%	6.9	39.6%	2.1%	9.7	55.8%	1,152
News Article	3.6%	5.6	86.6%	3.8%	5.3	82.5%	3.2%	6.2	95.5%	3,092
Personal Blog	3.6%	5.5	82.3%	3.7%	5.4	80.5%	3.4%	5.9	88.1%	2,974
Product Page	3.8%	5.2	89.8%	4.0%	5.1	87.0%	3.8%	5.3	91.9%	3,444
Weighted mean	2.6%	13.6	69.5%	2.7%	13.1	66.8%	2.2%	17.3	80.1%	20,000

Table 12: Per-class retrieval at 20% recall, Format. Columns as in Table 7.

H Full query results

Format	Harmonized		Raw		Topic	Harmonized		Raw	
	Prec.	Rec.	Prec.	Rec.		Prec.	Rec.	Prec.	Rec.
Tutorial	0.42	0.47	0.51	0.45	Finance & Business	0.58	0.45	0.67	0.36
Creative Writing	0.08	0.35	0.11	0.19	Home & Hobbies	0.69	0.23	0.72	0.20
News (Org.)	0.37	0.29	0.46	0.17	Transportation	0.60	0.38	0.65	0.19
About (Org.)	0.23	0.02	0.33	0.04	Politics	0.65	0.52	0.74	0.42
Structured Data	0.36	0.04	0.34	0.07	History	0.19	0.50	0.28	0.38
User Review	0.26	0.13	0.39	0.09	Literature	0.59	0.20	0.73	0.13
Personal Blog	0.46	0.68	0.57	0.61	Games	0.68	0.40	0.80	0.38
Content Listing	0.40	0.04	0.47	0.05	Art & Design	0.40	0.18	0.45	0.15
Knowledge Article	0.10	0.52	0.12	0.39	Science & Tech.	0.43	0.47	0.54	0.37
News Article	0.52	0.24	0.70	0.22	Crime & Law	0.64	0.45	0.76	0.41
Truncated	0.52	0.02	0.56	0.02	Health	0.52	0.68	0.60	0.62
FAQ	0.31	0.23	0.28	0.31	Entertainment	0.67	0.32	0.71	0.28
About (Pers.)	0.07	0.34	0.11	0.40	Education & Jobs	0.64	0.34	0.72	0.26
Product Page	0.60	0.27	0.64	0.28	Industrial	0.44	0.39	0.49	0.31
Spam / Ads	0.13	0.04	0.17	0.07	Fashion & Beauty	0.78	0.44	0.85	0.28
Nonfiction Writing	0.12	0.21	0.09	0.11	Software Dev.	0.53	0.37	0.63	0.30
Academic Writing	0.13	0.58	0.37	0.47	Travel	0.54	0.60	0.68	0.38
Q&A Forum	0.22	0.18	0.28	0.15	Adult	0.60	0.57	0.67	0.66
Comment Section	0.51	0.20	0.67	0.19	Social Life	0.39	0.23	0.39	0.21
Audio Transcript	0.41	0.32	0.47	0.25	Food & Dining	0.80	0.61	0.82	0.52
Documentation	0.07	0.77	0.09	0.67	Software	0.34	0.24	0.38	0.19
Legal Notices	0.30	0.53	0.33	0.46	Religion	0.48	0.71	0.52	0.62
Customer Support	0.07	0.32	0.14	0.24	Sports & Fitness	0.78	0.37	0.84	0.32
Listicle	0.54	0.06	0.57	0.05	Hardware	0.63	0.33	0.70	0.34

Class	Desc. (Harm.)	Desc. (Raw)	TF-IDF
edu	general context (35), development narratives (18), scientific-research (308), informing the reader (37), use of etymology (19), word history (26), holiday history (8), anti-bullying strategy (9), discovery and artifact context (7), teacher reflection (11)	scientific topic (53), educational resource (94), educational topic (45), historical account (212), scientific language (279), environmental topic (59), educational content (695), historical event (84), historical narrative (119), scientific study (143)	or (46,836), of the (53,488), species (1,197), century (2,786), of (75,396), scientists (1,195), researchers (1,416), nasa (345), used (14,612), children (6,776)
not-edu	product description (1,028), international travel guide (237), customer centricity (504), sports article (489), local news update (239), opinionated writing (196), local news (612), professional language use (762), tone - promotional (2,231), personal story (241)	home improvement (284), sports news (1,681), product description (2,078), varied topics (177), travel guide (473), financial (232), professional language (1,292), business news (493), respectful dialogue (92), customer-centric (1,279)	my (21,750), me (16,737), 2018 (2,622), 2014 (3,109), our (29,795), we (38,363), business (10,558), 2019 (2,062), am (10,094), 2016 (3,171)

Table 13: Top-10 features per class for FineWeb Edu, ranked by logistic regression coefficient, comparing harmonized descriptors, raw descriptors, and TF-IDF. Parenthesised numbers are training-set document frequencies. Descriptor tags shown; definitions omitted. Descriptors at min. frequency 1; TF-IDF capped at 50,000.