Tokenization of Internet Domains: Bridging NLP and Network Naming

Thomas Weiß and Timo Baumann
Department of Computer Science and Mathematics
Ostbayerische Technische Hochschule Regensburg
Seybothstraße 2, 93053 Regensburg, Germany
thomas.weiss@st.oth-regensburg.de
timo.baumann@oth-regensburg.de

Abstract

Language models require effective tokenization. Domain names, such as "aclanthology.org", represent a unique text genre with distinctive linguistic characteristics: they are very short, contain deliberate misspellings and abbreviations, and lack whitespace. These properties challenge tokenizers optimized for general text. We release a preprocessed domain-name corpus together with two new benchmark datasets for domain name analysis, and use them to compare domain-specific unigram and BPE tokenizers against tokenizers from pretrained language models (GPT, BERT, Llama, and others) on language classification (Common Crawl) and content classification (Curlie). On content classification, in-domain unigram tokenization outperforms the best general-purpose tokenizer by up to 5.6 percentage points in F1 micro and consistently surpasses in-domain BPE at every matched vocabulary size. On language classification, in-domain and general-purpose tokenizers perform comparably on F1 micro, though large-vocabulary LLM tokenizers (DeepSeek, Llama 3) can exceed in-domain results on F1 macro. These findings demonstrate that the benefit of domain-specific tokenization is task-dependent, with implications for domain recommendation, appraisal, and other computational linguistics tasks in the digital naming space.

1 Introduction

Domain names exhibit unique morphological characteristics due to the requirement that each domain name must be unique. This constraint encourages creative word formation: compound words (`facebook.com`), abbreviations (`ibm.com`), vowel deletions (`flickr.com`, `tumblr.com`), phonetic respellings (`reddit.com` from "read it"), and truncations that blend with the top-level domain (`bit.ly`, `instagr.am`).

These linguistic differences raise the question whether domain-specific tokenization can improve performance on Internet domain analysis tasks, since generic tokenizers may not capture the fine-grained morphological patterns present in domain names.

We conduct two classification experiments: language classification on domain names from the Common Crawl corpus, and content classification on the Curlie dataset. We compare unigram and BPE tokenizers, trained on Internet domain corpora, against the tokenizers shipped with widely used language models.

Our contributions are threefold: (1) a preprocessed domain corpus together with two new benchmark datasets for Internet domain analysis; (2) in-domain unigram and BPE tokenizers trained on this corpus; and (3) an empirical comparison showing that domain-specific unigram tokenization improves content-classification F1 micro by up to 5.6 percentage points (pp) over the best general-purpose baseline. For language classification, in-domain and general-purpose tokenizers perform comparably on F1 micro, while individual large-vocabulary LLM tokenizers exceed in-domain results on F1 macro.

2 Related Work

In contrast to domain names, conventional text documents targeted by language modeling typically consist of complete sentences with syntactic structure, word repetition, and complex semantic meaning. Spelling mistakes and grammatical errors are typically avoided or filtered from training material. While domain names are unique, word repetition leads to a highly biased vocabulary distribution (Zipf, 1935), known as Zipf's law.

Several tokenization approaches have been developed to address the open-vocabulary problem. Byte-Pair Encoding (BPE, Gage, 1994; Sennrich et al., 2016) iteratively merges the most frequent character pairs, while unigram language modeling

(Kudo, 2018) uses a probabilistic approach that can produce multiple segmentations. SentencePiece (Kudo and Richardson, 2018) provides a unified framework implementing both and enables training directly from raw text. Bostrom and Durrett (2020) compared BPE and unigram tokenization for language model pretraining, finding that unigram produces morphologically more coherent subword units and matches or outperforms BPE across downstream tasks, suggesting that tokenization choice impacts model performance.

Language identification for short text snippets presents particular challenges due to limited context. Toftrup et al. (2021) reproduced a bidirectional LSTM previously described in an Apple blog post (Apple, 2019) for language identification in short strings, demonstrating that specialized architectures can outperform general-purpose identifiers on very short text.

An alternative approach to subword tokenization is to operate directly on character or byte sequences, bypassing tokenization. Pagnoni et al. (2025) recently proposed the Byte Latent Transformer, which processes text as byte patches and demonstrates that byte-level processing can scale effectively. These tokenizer-free approaches may be particularly suitable for domain names, where character-level patterns are crucial and the lack of whitespace complicates subword segmentation.

Recent research has examined the influence of tokenization on LLM performance. Dagan et al. (2024) showed that adapting tokenization to a specific domain yields gains in generation speed and effective context size without performance loss. Edman et al. (2024) introduced the CUTE benchmark for evaluating LLMs' character-level understanding of their own tokens, finding that models understand token composition but largely fail to grasp orthographic similarity.

Despite these advances, tokenization strategies specifically for domain names remain underexplored. We address this gap by comparing in-domain trained tokenizers against generic LLM tokenizers on two domain name classification tasks and releasing benchmark datasets for future work.

3 Datasets

We use three datasets, each serving a distinct role. From the *Domains Project* (DomainsProject, 2025), we derive a broad collection of registered domain names (Weiß and Baumann, 2026c) used exclusively to train our domain tokenizers. From the *Common Crawl Index* (Common Crawl Foundation, 2026), we derive our benchmark for language classification (Weiß and Baumann, 2026a), while from the *Curlie* web directory (Lugeon and Piccardi, 2022), we derive a benchmark for content classification (Weiß and Baumann, 2026b). All three derived datasets are publicly released on Zenodo to support reproducibility and future work.

All datasets undergo the same validation pipeline. Domains are converted to their Unicode (NFC normalized) form following IDNA 2008 (Klensin, 2010) with UTS #46 compatibility processing in non-transitional mode. Domains registered under Public Suffix List (PSL) private suffixes are removed, as they represent machine-generated or user-provisioned subdomains under shared platforms rather than independently registered names. The filtered entries are dominated by reverse-DNS hostnames of cloud virtual machines (e.g. `*.compute.amazonaws.com` across multiple AWS regions), blogging and e-commerce platforms (`blogspot.com`, `myshopify.com`), developer hosting services (`herokuapp.com`, `github.io`, `vercel.app`), and consumer dynamic DNS and NAS services (`myfritz.net`, `synology.me`). Entries that fail IDNA 2008 validation are also discarded, even though some resolve in practice. An example is the emoji domain `xn--i-7iq.ws`, which browsers might render as `i♥.ws`. Across all three corpora, over 99 % of domains are pure ASCII. The remainder are valid Internationalized Domain Names whose non-ASCII characters directly encode script and language information. Table 1 summarizes the preprocessing pipeline for all three datasets.

3.1 Domains Project: Tokenizer Training

We train our in-domain tokenizers on domain names from the Domains Project (DomainsProject, 2025), the world's largest publicly available dataset of Internet domain names. We deliberately separate the tokenizer training data from the downstream evaluation datasets to avoid biasing the learned vocabulary toward the specific domain distributions of either classification task. Because all three corpora sample from the same global domain name space, some overlap is inevitable: 24.1 M domains are shared with Common Crawl and 783 K with Curlie; only 463 K appear in all three datasets. Nevertheless, 246.3 M of the 270.8 M training domains (91.0 %) occur in neither evaluation corpus, so the

Step	Domains Project	Common Crawl	Curlie
URLs		2,791,068,710	
hostnames	1,764,770,720	64,647,914	885,582
Filtered (IP address)	0	88,214	6
Filtered (suffix only)	646	1,158	
Filtered (malformed)	17,448	35	
Collapsed to apex domain	1,427,894,341	29,508,573	60,597
Unique registerable domains	336,858,285	35,049,969	824,979
Filtered (PSL private)	66,085,596	530,983	11,055
Filtered (IDNA 2008)	9,966	38	
Unique valid domains		34,518,948	813,924
Task-specific filtering		156,909	4,172
Final corpus	270,762,723	34,362,039	809,752

Table 1: Preprocessing results for all three corpora. Empty cells indicate steps not applicable to a given dataset.

learned vocabulary is not likely to be driven by the evaluation distributions.

The subdomain distribution is heavily skewed, with a few ISP domains accounting for hundreds of millions of subdomains (Appendix A.1).

Of the 270.8 million domains in the tokenizer training corpus, 269.1 M (99.4 %) are ASCII and 1.6 M are IDNs. The 9,966 IDNA-filtered entries contain emojis (8,877), IDNA 2008 encoding failures (808), contain underscores (275), or invisible characters (6).

The training corpus spans 1,429 unique TLDs (out of 1,590 IANA-registered TLDs). By TLD type, 68.3 % of domains fall under generic TLDs and 30.9 % under country-code TLDs, with sponsored (0.2 %) and other types making up the remainder. `.com` alone represents 51.4 % of all domains (139.1M), followed by `.net` (4.5 %), `.de` (4.2 %), `.org` (3.7 %), and `.uk` (2.0 %).

3.2 Common Crawl: Language Classification

We use the index data from the Common Crawl project CC-MAIN-2026-04[1], which provides URLs labeled by languages as classified by CLD2[2]. CLD2 is a Naive Bayes classifier operating on the HTML text content of each crawled web page, not the domain name string itself. A recent study found that CLD2 offers the best trade-off between coverage and inference speed among current language identification tools (Suarez et al., 2026). We use these page-level language labels as proxy labels for the corresponding domain names.

For each valid URL, we aggregate page-level CLD2 detections across all pages sharing the same apex domain, producing a language-proportion vector per domain. We then apply a threshold $\tau = 0.1$: any language with a proportion above τ becomes a positive label, casting domain-language assignment as a multi-label classification task. 156,909 domains where no language exceeds τ are discarded.

Of the 161 language labels discovered in the index, many occur fewer than a few thousand times and cannot support reliable evaluation after splitting. We retain the 30 languages with at least 100 000 domain-language occurrences. This threshold aligns with a natural gap in the frequency distribution (the 30th-ranked language, Hebrew, has 106 K occurrences versus 82 K for the next) and ensures that every class has $>$10 K test samples after a 70/20/10 split.

The resulting label set spans eight script families and includes 20 Latin script, 3 Cyrillic, 2 CJK, and one each of Arabic, Greek, Hangul, Hebrew, and Thai script languages (Appendix A.3). The maximum imbalance ratio is $\sim$237 (English 25.1 M vs. Hebrew 106 K), which is steep but tractable.

The per-language composition is detailed in Table 2. The majority of domains carry a single language label, with an average label cardinality of 1.36 (Appendix A.2). English alone accounts for 73.1 % of all labels. The final dataset contains 34,362,039 trainable domains over 30 language classes (Table 1).

[1] https://data.commoncrawl.org/crawl-data/CC-MAIN-2026-04/index.html (accessed: 14.02.2026)

[2] See https://commoncrawl.github.io/cc-crawl-statistics/plots/languages (accessed: 14.02.2026)

Lang	Domains	%	Lang	Domains	%
eng	25,100,666	73.05	ces	379,680	1.10
deu	3,327,935	9.68	swe	301,628	0.88
zho	2,654,024	7.72	kor	289,573	0.84
jpn	2,536,128	7.38	vie	280,515	0.82
fra	2,068,910	6.02	hun	176,402	0.51
spa	1,600,880	4.66	ara	161,216	0.47
rus	1,316,615	3.83	ell	156,889	0.46
nld	978,614	2.85	tha	151,102	0.44
ita	868,195	2.53	srp	149,086	0.43
lat	815,813	2.37	fin	140,139	0.41
por	751,436	2.19	ron	134,476	0.39
pol	599,296	1.74	slk	131,507	0.38
dan	485,826	1.41	nor	130,172	0.38
ind	423,909	1.23	ukr	127,040	0.37
tur	404,312	1.18	heb	106,095	0.31

Table 2: Per-language counts in the Common Crawl language dataset ($\tau = 0.1$, multi-label, 30 selected languages, imbalance 237:1). Proportions sum to more than 100 % because a domain can carry multiple labels.

Category	Count	%
Business	246,654	30.5
Society	113,505	14.0
Arts	72,273	8.9
Recreation	69,067	8.5
Shopping	67,492	8.3
Sports	56,925	7.0
Health	50,446	6.2
Computers	49,165	6.1
Science	32,437	4.0
Reference	25,629	3.2
Home	12,041	1.5
Games	12,001	1.5
News	9,335	1.2
Kids_and_Teens	8,496	1.1

Table 3: Category distribution of the *Curlie* dataset across 14 content classes (imbalance 29:1). Proportions sum to more than 100 % because 15,200 domains (1.9 %) carry two or more labels.

3.3 Curlie: Content Classification

The *Curlie* dataset provides a complementary benchmark for content classification of domain names. Unlike language classification, content categories are less directly reflected in the domain string itself but may stem from their meaning.

We derive the dataset from the Curlie web directory[3], the successor to the Open Directory Project (DMOZ), using the processed version published by Lugeon et al. (2022). The top-level categories yield 14 content classes. From the 885,582 labeled domain names in that project, we collapse each to its apex domain and retain the union of all associated category labels. We remove apex domains with contradictory label assignments as well as entries on PSL-private suffixes or failing IDNA 2008 validation, yielding 809,752 domains in the final dataset (Table 1).

As shown in Table 3, the dataset exhibits substantial class imbalance: *Business* accounts for 30.5 % while *Kids_and_Teens* represents just 1.1 % (ratio 29:1), comparable to the English dominance in the Common Crawl dataset. TLDs may carry clear content signal. For instance, `.edu` maps almost exclusively to *Reference* (84.9 %) while generic TLDs show a broader category mix (Appendix A.4).

4 Tokenizers

We train in-domain BPE (Gage, 1994) and unigram (Kudo, 2018) tokenizers on the Domains Project corpus described in Section 3.1, using the SentencePiece library (Kudo and Richardson, 2018). We experiment with vocabulary sizes from 1,000 to 100,000 tokens (Table 5). For reference, note that GPT-4's 100,276 size vocabulary splits into 45,962 space-prefixed and 54,314 non-space-prefixed tokens. This is relevant because domain names contain no whitespace.

We implement a unified wrapper interface for plug-and-play comparison of tokenizers shipped with publicly available LLMs. The selected tokenizers (Table 5) cover the three main tokenization families: BPE (GPT-4 (OpenAI, 2022), DeepSeek (DeepSeek-AI et al., 2024), Llama 3 (Grattafiori et al., 2024), Llama 2 (Touvron et al., 2023), RoBERTa (Liu et al., 2019), BLOOM (BigScience Workshop et al., 2022), Mistral Tekken (Jiang et al., 2023)), Unigram (BigBird (Zaheer et al., 2020), T5 (Raffel et al., 2020), ALBERT (Lan et al., 2020), XLNet (Yang et al., 2019)), and WordPiece (BERT, mBERT (Devlin et al., 2019)). While not exhaustive, our selection is representative of the tokenization approaches currently in use.

As a baseline, we include a Unicode character-level tokenizer, yielding 12,439 tokens. This represents the most fine-grained segmentation above the byte level.

All tokenizers are applied to the full domain string without special treatment of the top-level domain (TLD).

Qualitatively, our domain-specific tokenizers produce more linguistically meaningful subword units, as illustrated by the tokenization of `smokeheadguitars.co.uk` (Table 4). Tokenizers from pre-trained models often produce splits that cross morphological boundaries.

[3] `https://curlie.org/` (accessed: 14.02.2026)

General-purpose (BPE)	
GPT-4:	sm oke head g uit ars .co .uk
+ws:	␣smoke head g uit ars .co .uk
DeepSeek:	sm oke head g uit ars .co .uk
+ws:	␣smoke head g uit ars .co .uk
Llama 3:	sm oke head g uit ars .co .uk
+ws:	␣smoke head g uit ars .co .uk
Llama 2:	smoke head gu it ars . co . uk
+ws:	␣smoke head gu it ars . co . uk
RoBERTa:	sm oke head g uit ars . co . uk
+ws:	␣smoke head g uit ars . co . uk
BLOOM:	sm oke head gu itar s . co . uk
+ws:	␣smoke head gu itar s . co . uk
M. Tekken:	sm oke head gu it ars .co .uk
+ws:	␣smoke head gu it ars .co .uk
General-purpose (Unigram)	
BigBird:	smoke head gu itars . co . uk
T5:	smoke head gui tar s . co . uk
ALBERT:	smoke head guitar s . co . uk
XLNet:	smoke head guitar s . co . uk
General-purpose (WordPiece)	
BERT:	smoke head gui tar s . co . uk
mBERT:	smoke head guita rs . co . uk
In-domain unigram (ours)	
unigram-1K:	s mo ke he ad gu it ar s . co . uk
unigram-5K:	smoke head guitar s . co . uk
unigram-10K:	smoke head guitar s . co . uk
unigram-30K:	smoke head guitars . co . uk
unigram-50K:	smoke head guitars . co . uk
unigram-100K:	smoke head guitars . co . uk
In-domain BPE (ours)	
BPE-1K:	s mo ke he ad gu it ar s . co . uk
BPE-5K:	smo ke head gu it ars . co . uk
BPE-10K:	smoke head guitars . co . uk
BPE-30K:	smoke head guitars . co . uk
BPE-50K:	smoke head guitars . co . uk
BPE-100K:	smoke head guitars . co . uk
Baseline char:	smokeheadguitars.co.uk
Morphology :	smoke head guitars .co .uk

Table 4: Tokenization of smokeheadguitars.co.uk. Rows marked + *ws* prepend a leading whitespace.

Likewise, Bostrom and Durrett (2020) demonstrate that unigram tokenization produces morphologically more coherent splits compared to BPE tokenization, which is consistent with our observations. Notably, the ALBERT and XLNet tokenizers (both unigram-based) also produce relatively good morphological splits, providing supporting evidence for the unigram advantage.

An additional German-language example confirms the same trend (Appendix B).

4.1 Tokenizer Normalization

To compare tokenizers fairly, we strip tokenizer-specific artifacts (WordPiece ## prefixes, SentencePiece _ boundary markers and other special tokens) so that metrics reflect only actual text content. The full normalization details are described in Appendix C.

Several byte-level BPE tokenizers (GPT-4, Llama 3, DeepSeek, BLOOM, RoBERTa, Mistral Tekken) are sensitive to a leading whitespace character, which changes the first token they emit. Both settings are compared on the best-performing architecture per task: BiLSTM for content classification and the Decoder for language classification. The effect is small ($<$0.4 pp) and varies by task: omitting the space tends to benefit BiLSTM, whereas the Decoder shows mixed results depending on the tokenizer. As the differences are marginal, we omit the leading space throughout the final comparison.

4.2 Tokenizer Comparison

Table 5 reports vocabulary utilization for all tokenizers on the Common Crawl Index and Curlie datasets, showing the fraction of vocabulary tokens actually activated on domain name data.

In-domain tokenizers achieve near-complete vocabulary utilization ($>$99 % on Common Crawl), while general-purpose tokenizers activate only a small fraction of their vocabulary on domain data. Since embedding parameters scale linearly with vocabulary size, this translates into substantial parameter overhead for large-vocabulary models (Table 5).

5 Model

We compare four architectures, all trained from scratch without pretrained weights:

BiLSTM – a 2-layer bidirectional LSTM following Toftrup et al. (2021), originally designed for short-text language identification.

Encoder – a 2-layer encoder-only Transformer (Vaswani et al., 2017) with masked mean pooling (Reimers and Gurevych, 2019) over non-padding positions.

Decoder – a 2-layer decoder-only (GPT-style) Transformer (Radford et al., 2018) that classifies from the last non-padding token.

Transformer – a 2-layer encoder–decoder Transformer that classifies from the last decoder token.

All four share a 512-dimensional embedding, a 768-dimensional feed-forward hidden layer, and dropout of 0.1. The attention-based variants additionally use 8 attention heads of 64 dimensions each and learned absolute positional embeddings. Training uses AdamW for up to 50 epochs with an

Tokenizer	#tok	Used (%) Common Crawl	Curlie
General-purpose (BPE)			
GPT-4	100,276	22,410 (22.4)	17,309 (17.3)
DeepSeek	128,815	30,141 (23.4)	16,174 (12.6)
Llama 3	128,256	30,111 (23.5)	17,885 (13.9)
BLOOM	250,680	31,896 (12.7)	18,496 (7.4)
RoBERTa	50,265	11,874 (23.6)	10,515 (20.9)
Llama 2	32,000	21,410 (66.9)	14,877 (46.5)
Mistral Tekken	131,072	28,648 (21.9)	16,227 (12.4)
General-purpose (Unigram)			
XLNet	32,000	20,097 (62.8)	15,071 (47.1)
BigBird	50,358	30,988 (61.5)	23,077 (45.8)
ALBERT	30,000	28,950 (96.5)	23,523 (78.4)
T5	32,100	20,528 (64.0)	15,301 (47.7)
General-purpose (WordPiece)			
BERT	30,522	28,001 (91.7)	21,824 (71.5)
mBERT	105,879	64,049 (60.5)	37,177 (35.1)
In-domain unigram (ours)			
unigram-1K	1,000	998 (99.80)	784 (78.40)
unigram-5K	5,000	4,998 (99.96)	4,774 (95.48)
unigram-10K	10,000	9,985 (99.85)	9,677 (96.77)
unigram-30K	30,000	29,951 (99.84)	28,246 (94.15)
unigram-50K	50,000	49,898 (99.80)	45,438 (90.88)
unigram-100K	100,000	99,849 (99.85)	83,273 (83.27)
In-domain BPE (ours)			
BPE-1K	1,000	998 (99.80)	784 (78.4)
BPE-5K	5,000	4,998 (99.96)	4,782 (95.6)
BPE-10K	10,000	9,991 (99.91)	9,739 (97.4)
BPE-30K	30,000	29,967 (99.89)	28,838 (96.1)
BPE-50K	50,000	49,936 (99.87)	46,614 (93.2)
BPE-100K	100,000	99,828 (99.83)	85,082 (85.1)
Baseline			
char	12,439	5,396 (43.4)	80 (0.6)

Table 5: Vocabulary utilization on the Common Crawl and Curlie corpora. #tok is the total vocabulary size; Used columns show activated token count (%).

effective batch size of 4 096 and a learning rate of 3×10^{-4} (1×10^{-3} for the BiLSTM). All models are trained with *Focal Loss* on per-label sigmoid logits, which down-weights well-classified examples and provides better gradients than plain cross-entropy under heavy label imbalance. Checkpoints are saved for the best validation performance on each tracked metric. Reported test F1 micro and F1 macro therefore come from different checkpoints of the same run. All settings are held fixed so that the tokenizer is the only experimental variable. Table 6 summarizes the configurations. Further architecture details are in Appendix D.

6 Experiments

We evaluate all tokenizers from Table 5 across the four architectures of Section 5 on language classification (Section 6.1) and content classification

Architecture	Emb. Dim	Hidden Dim	Layers	Heads
BiLSTM	512	768	2	–
Encoder	512	768	2	8
Transformer	512	768	2	8
Decoder	512	768	2	8

Table 6: Model architecture summary. All models are trained from scratch without pretrained weights.

Architecture	Sel. by	Tokenizer	F1 Micro	F1 Macro
Best in-domain				
Decoder	F1 micro	unigram-100K	**0.762**	0.514
	F1 macro	BPE-50K	0.758	**0.580**
Transformer	F1 micro	unigram-30K	0.761	0.493
	F1 macro	BPE-1K	0.760	0.543
Encoder	F1 micro	unigram-100K	0.758	0.468
BiLSTM	F1 micro	unigram-50K	0.756	0.440
Best general-purpose				
Decoder	both	DeepSeek	**0.760**	**0.588**
Transformer	both	DeepSeek	**0.756**	**0.521**
Encoder	both	DeepSeek	**0.753**	**0.535**
BiLSTM	F1 micro	GPT-4	**0.757**	0.437
	F1 macro	T5	0.754	**0.439**
Baseline		ccTLD heuristic	0.717	0.197

Table 7: Language classification: best tokenizer per architecture on the test set.

(Section 6.2). Hyperparameters are held constant so that any variation is due to the tokenizer, the architecture, or random initialization.

Both tasks are framed as multi-label classification. We report F1 micro and F1 macro (Sokolova and Lapalme, 2009) from separate best-validation checkpoints (Section 5), so neither metric is penalized by a selection rule tuned for the other.

6.1 Language Classification

Using the Common Crawl language dataset (Section 3.2), we treat language classification as a multi-label task over the 30 languages that exceed the 100 K-occurrence threshold.

As a non-neural baseline, we map country-code TLDs (ccTLDs) to their associated languages (`.de` → deu, `.fr` → fra, etc.) and default to English for generic TLDs such as `.com` and `.net` (Appendix A.3, Table 13). This heuristic quantifies the predictive signal that resides in the TLD alone and provides a lower bound for the neural models.

The best in-domain results exceed the ccTLD baseline by +4.5 pp on F1 micro (0.762 vs. 0.717) and by +38.3 pp on F1 macro (0.580 vs. 0.197). The small micro margin partly reflects English's

73 % share of labels. F1 macro better captures the per-language gains. Across architectures, the best-per-architecture F1 micro scores span only 0.6 pp (BiLSTM 0.756 to Decoder 0.762), whereas F1 macro spans ∼14 pp (best by macro: BiLSTM 0.440 to Decoder 0.580) with a consistent ordering Decoder > Transformer > Encoder > BiLSTM (Table 7).

Selecting by F1 micro yields a unigram tokenizer on every architecture. Selecting by F1 macro instead promotes BPE on the Decoder (BPE-50K, +6.6 pp macro) and the Transformer (BPE-1K, +5.0 pp macro). For the Encoder and BiLSTM the two selections differ by less than 0.1 pp on macro.

General-purpose tokenizers are competitive on F1 micro across all four architectures (within 0.5 pp of in-domain) but diverge on F1 macro (Table 7). On the Decoder, DeepSeek reaches F1 micro 0.7595 (within 0.2 pp of the in-domain best) and F1 macro 0.5878, *exceeding* the best in-domain result (BPE-50K, 0.5795) by +0.8 pp; Llama 3 follows closely at 0.5862 (+0.7 pp). On the Encoder the gap widens: DeepSeek attains F1 macro 0.5354, +6.6 pp above the best in-domain tokenizer (unigram-1K, 0.4692), with Llama 3 (+6.3 pp) and Mistral Tekken (+4.2 pp) also exceeding the in-domain best. On the Transformer and BiLSTM, no general-purpose tokenizer exceeds the in-domain macro. DeepSeek reaches 0.5209 (within 2.2 pp of in-domain BPE-1K on the Transformer), and GPT-4 matches in-domain on BiLSTM micro within 0.1 pp. The macro advantage is thus limited to the large-vocabulary tokenizers from recent multilingual LLMs (DeepSeek 128 K, Llama 3 128 K, Mistral Tekken 131 K).

Vocabulary size has little effect on the language classification task. Across 1 K–100 K, in-domain unigram F1 micro varies by only 0.2–0.7 pp on every architecture (BiLSTM 0.2 pp, Encoder 0.6 pp, Transformer 0.6 pp, Decoder 0.7 pp), an order of magnitude smaller than the +9.3 pp range observed on the Transformer for content over the same vocabulary span.

6.2 Content Classification

Content classification poses a qualitatively different challenge. Whereas language leaves direct traces in the character sequences and TLD choices of a domain, the mapping from a domain name to its content category is considerably more difficult. We use the Curlie dataset (Section 3.3) with 14 content categories as a multi-label task.

Architecture	Tokenizer	F1 Micro	F1 Macro
Best in-domain			
BiLSTM	unigram-30K	0.512	0.417
Transformer	unigram-50K	0.427	0.389
Encoder	unigram-100K	0.388	0.359
Decoder	unigram-50K	0.372	0.347
Best general-purpose			
BiLSTM	XLNet	0.499	0.398
Transformer	ALBERT	0.378	0.347
Encoder	ALBERT	0.333	0.310
Decoder	ALBERT	0.316	0.297
Character only			
BiLSTM	char	0.495	0.402
Transformer	char	0.127	0.128
Encoder	char	0.104	0.096
Decoder	char	0.149	0.151
Baseline	TLD heuristic	0.343	0.076

Table 8: Content classification: best tokenizer per architecture on the test set.

As a non-neural baseline, we assign each TLD the content category most frequently associated with it in the training data. Domains with unseen TLDs default to the globally most frequent category. Because most TLDs cover a broad category mix rather than concentrating on a single class, this baseline is expected to be weaker than the TLD-to-language heuristic. Empirically it reaches F1 micro 0.343 and F1 macro 0.076, less than half the language ccTLD macro (0.197), consistent with the broad category mix observed across most TLDs (Appendix A.4).

Every neural model with a subword tokenizer at least triples the TLD baseline on F1 macro (lowest 0.246 vs. 0.076), confirming that the second-level domain carries content signal beyond the TLD. On F1 micro, the Encoder and Decoder paired with the general-purpose ALBERT tokenizer (0.333 and 0.316) fall below the TLD baseline (0.343), whereas the best in-domain tokenizer exceeds it on every architecture.

Architecture has a pronounced effect on F1 macro. The ranking is BiLSTM > Transformer > Encoder > Decoder, each paired with an in-domain unigram tokenizer (Table 8). This inverts the language ranking, where the Decoder is strongest (Section 6.1).

The in-domain advantage on F1 micro is largest on the Decoder (+5.6 pp over ALBERT, 17.6 % relative), followed by +5.5 pp on the Encoder, +4.9 pp on the Transformer, and +1.3 pp on the BiLSTM. The unigram-based general-purpose tokenizers (XLNet, ALBERT, T5) consis-

Architecture	1 K	10 K	30 K	50 K	100 K
BiLSTM	0.500	0.511	**0.512**	0.511	0.511
Transformer	0.334	0.408	0.423	**0.427**	0.427
Encoder	0.271	0.360	0.378	0.386	**0.388**
Decoder	0.276	0.347	0.367	**0.372**	0.370

Table 9: Content classification: in-domain unigram F1 micro by architecture and vocabulary size.

tently outperform the BPE-based LLM tokenizers (`mistral-tekken`, `facebook-llama3`, `deepseek`, `openai-gpt4`) across all architectures.

In-domain unigram outperforms in-domain BPE at every matched vocabulary size on every architecture (e.g. on the Transformer at 30 K, +4.5 pp: 0.423 vs. 0.378). The effect of vocabulary size, by contrast, is strongly architecture-dependent (Table 9): BiLSTM scores remain nearly constant (1.2 pp across 1 K–100 K), whereas the attention-based architectures gain +9 to +12 pp from 1 K up to a peak at 50 K–100 K before plateauing.

The character tokenizer produces the largest gap between architectures (Table 8, *Character only*). The BiLSTM reaches F1 micro 0.495, within 0.4 pp of XLNet and above every BPE- and WordPiece-based variant on this architecture, whereas the attention-based architectures fall below 0.15.

7 Conclusion and Future Work

We present a systematic comparison of domain-specific and general-purpose tokenization for domain name classification across two tasks and four architectures.

For language classification, every neural model exceeds the ccTLD baseline, with the best in-domain tokenizer improving F1 micro by +4.5 pp and F1 macro by +38.3 pp. Architecture choice has only a minor effect on F1 micro (0.6 pp spread) but strongly influences F1 macro (∼14 pp spread), where the Decoder performs best. On F1 micro, general-purpose tokenizers perform comparably to in-domain ones. On F1 macro, the large-vocabulary LLM tokenizers DeepSeek and Llama 3 exceed the best in-domain result on the Decoder (+0.8 pp) and Encoder (+6.6 pp). Other general-purpose tokenizers do not show this advantage.

For content classification, the in-domain advantage is more pronounced: up to +5.6 pp F1 micro over the best general-purpose tokenizer (ALBERT on the Decoder). The architecture ordering inverts relative to language, with BiLSTM performing best. On this task, in-domain unigram outperforms in-domain BPE at every matched vocabulary size on every architecture, confirming Bostrom and Durrett (2020) in a new, whitespace-free text genre. Vocabulary scaling is strongly architecture dependent. The attention-based models gain +9 to +12 pp from 1 K to their peak at 50 K–100 K, whereas BiLSTM remains nearly constant, enabling smaller vocabularies without sacrificing performance.

We additionally release a preprocessed corpus and two new benchmarks to support future work on Internet domain analysis: the 270.8 M-domain Domains Project corpus used to train our tokenizers, the 34.4 M-domain Common Crawl multi-label language classification benchmark, and the 810 K-domain Curlie multi-label content classification benchmark.

Beyond the classification tasks studied here, domain-specific tokenization may benefit related applications such as domain recommendation and appraisal, where understanding the semantic content of a domain name is relevant. The trained models, with further refinement, may also support web crawling, domain parking, and broader web analytics. Additionally, a reevaluation of content classification taxonomies for the domain name context could yield further improvements. Current taxonomies are designed for general web content and may not optimally capture the nuances specific to domain names.

Acknowledgements

We thank Tobias Mauerer for providing valuable comments on an early version of this manuscript and OpusDNS for supporting this work. We also thank the anonymous reviewers for their constructive feedback.

References

Apple. 2019. Language identification from very short strings. Apple Machine Learning Research. Accessed: 2026-05-09.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, and 180 others. 2022. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. *arXiv e-prints*, arXiv:2211.05100.

Kaj Bostrom and Greg Durrett. 2020. Byte pair encoding is suboptimal for language model pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

Common Crawl Foundation. 2026. Common Crawl index. https://commoncrawl.org/. Accessed: 2026-02-14.

Gautier Dagan, Gabriel Synnaeve, and Baptiste Rozière. 2024. Getting the most out of your tokenizer for pre-training and domain adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2024. DeepSeek-V3 Technical Report. *arXiv e-prints*, arXiv:2412.19437.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

DomainsProject. 2025. DomainsProject.org – world's largest internet domains dataset. Over 2.975 billion domains indexed across 1,251 TLDs. BSD-3-Clause licensed.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' understanding of their tokens. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3017–3026, Miami, Florida, USA. Association for Computational Linguistics.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, 12(2):23–38.

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, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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

John C. Klensin. 2010. Internationalized Domain Names in Applications (IDNA): Protocol. RFC 5891.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *Preprint*, arXiv:1907.11692.

Sylvain Lugeon and Tiziano Piccardi. 2022. Curlie dataset – language-agnostic website embedding and classification.

Sylvain Lugeon, Tiziano Piccardi, and Robert West. 2022. Homepage2Vec: Language-agnostic website embedding and classification. In *Proceedings of the Sixteenth International AAAI Conference on Web and Social Media, ICWSM 2022, Atlanta, Georgia, USA, June 6-9, 2022*, pages 1285–1291. AAAI Press.

OpenAI. 2022. tiktoken - fast BPE tokeniser for use with OpenAI's models. https://github.com/openai/tiktoken. (visited 2024-11-09).

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. Byte latent transformer: Patches scale better than tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. Technical report, OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text

transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Pedro Ortiz Suarez, Laurie Burchell, Catherine Arnett, Rafael Mosquera, Sara Hincapié Monsalve, Thom Vaughan, Damian Stewart, Malte Ostendorff, Idris Abdulmumin, Vukosi Marivate, Shamsuddeen Hassan Muhammad, Atnafu Lambebo Tonja, Hend Al-Khalifa, Nadia Ghezaiel Hammouda, Verrah Akinyi Otiende, Tack Hwa Wong, Jakhongir Saydaliev, Melika Nobakhtian, Muhammad Ravi Shulthan Habibi, and 78 others. 2026. CommonLID: Re-evaluating state-of-the-art language identification performance on web data. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 33063–33080, San Diego, California, United States. Association for Computational Linguistics.

Mads Toftrup, Søren Asger Sørensen, Manuel R. Ciosici, and Ira Assent. 2021. A reproduction of Apple's bi-directional LSTM models for language identification in short strings. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 36–42, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Weiß and Timo Baumann. 2026a. Common Crawl domain-level multi-label language identification benchmark. Dataset on Zenodo. DOI: 10.5281/zenodo.19943679.

Thomas Weiß and Timo Baumann. 2026b. Curlie domain-level multi-label content classification benchmark. Dataset on Zenodo. DOI: 10.5281/zenodo.19943663.

Thomas Weiß and Timo Baumann. 2026c. Preprocessed apex domains from the Domains Project for tokenizer training. Dataset on Zenodo. DOI: 10.5281/zenodo.19831167.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big Bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

George Kingsley Zipf. 1935. *The Psycho-Biology of Language: An Introduction to Dynamic Philology*. Houghton Mifflin, Oxford, England.

A Additional Dataset Statistics

A.1 Most Collapsed Subdomains in the Domains Project

Table 10 lists the ten registerable domains with the highest subdomain occurrence counts in the raw Domains Project data. The list is dominated by large Internet service providers and telecommunications carriers whose infrastructure spans millions of reverse-DNS hostnames. `comcast.net` alone accounts for 87.2 million entries, followed by `sbcglobal.net` (63.4M) and `myvzw.com` (57.7M). The cloud infrastructure provider `amazonaws.com` appears at rank 7 with 30.0 million subdomains. Together, these ten domains contribute 457.8 million entries (13.2 % of all entries) illustrating how a handful of heavily subdivided networks would dominate the corpus without collapsing subdomains.

Rank	Domain	Subdomains
1	`comcast.net`	87,242,985
2	`sbcglobal.net`	63,382,609
3	`myvzw.com`	57,712,007
4	`rr.com`	47,555,645
5	`hinet.net`	44,559,735
6	`spectrum.com`	42,165,773
7	`amazonaws.com`	30,014,547
8	`virginm.net`	29,553,653
9	`sfr.net`	28,788,331
10	`verizon.net`	26,774,839

Table 10: Top 10 registerable domains by occurrence count in the raw Domains Project data. Counts reflect the number of distinct subdomains observed for each domain.

A.2 Multi-Label Statistics for Common Crawl

Table 11 details the label-set size distribution and top language co-occurrences for the Common Crawl language dataset. The majority of domains carry a single language label (average label cardinality 1.36). The most frequent pair is *deu* + *eng*, followed by *eng* + *jpn*. Regarding language confidence, the vast majority of proxy labels are high-confidence, with most domains having a dominant-language proportion ≥ 90 % of all URLs per apex domain.

(a) Label-set size

Labels	Domains	%
1	23,227,720	67.6
2	9,996,805	29.1
3	1,041,134	3.0
4	81,843	0.2
5	11,883	<0.1
6	2,180	<0.1
7	344	<0.1
8+	130	<0.1

(b) Top co-occurrences

Pair	Domains	%
deu + eng	1,580,196	4.6
eng + jpn	1,452,729	4.2
eng + fra	1,256,422	3.7
eng + zho	934,685	2.7
eng + spa	878,274	2.6
eng + lat	710,495	2.1
eng + nld	617,239	1.8
eng + ita	610,863	1.8
eng + rus	546,747	1.6
jpn + zho	526,985	1.5

Table 11: Multi-label statistics for the Common Crawl language dataset (τ=0.10, 30 languages): (a) label-set size distribution, and (b) top language co-occurrences.

A.3 Language Selection

Of the 160 languages classified by CLD2 in the Common Crawl index, domain-language counts follow a heavy-tailed distribution. Table 12 compares four frequency-based cutoffs derived from natural gaps in this distribution. We select Cut B (>100 K, 30 languages). This is a round threshold that aligns with the gap after rank 30 (heb 106 K $\rightarrow$ cat 82 K, -23 %) and keeps the imbalance ratio at $\sim$237$\times$. It provides ccTLD coverage for 29 of the 30 classes. The selected languages span eight script families and are listed in Table 13.

Cut	Langs	Thresh.	Gap	Max imb.	Min test
A	18	>280 K	-37 %	$\sim$90$\times$	$\sim$28 K
B	**30**	**>100 K**	**-23 %**	**$\sim$237$\times$**	**$\sim$11 K**
C	37	>44 K	-29 %	$\sim$568$\times$	$\sim$4.4 K
D	43	>20 K	-30 %	$\sim$1,155$\times$	$\sim$2.2 K

Table 12: Language-count cutoff alternatives. The gap column shows the percentage drop from the last included language to the first excluded one. Max imb. is the size ratio between the largest and smallest class; Min test is the smallest per-class test split (10 %).

Note on Latin (`lat`). Latin ranks 10th by count with 815 813 occurrences despite having no living native-speaker community and no ccTLD. We retain `lat` because it meets the frequency threshold.

A.4 Per-TLD Content Profiles in Curlie

Table 14 reports per-TLD category distributions for the 15 largest TLDs in the Curlie dataset plus `.info` and `.edu` as notable generic and purpose-specific TLDs. Category distributions vary remarkably. For example, `.org` domains are predominantly *Society* (43.4 %), `.edu` domains are almost

# Code / Language	Count	Script	ccTLD coverage
1 ara – Arabic	161,216	Arabic	.sa, .ae, .eg, .jo, .iq, …
2 ces – Czech	379,680	Latin	.cz
3 dan – Danish	485,826	Latin	.dk
4 deu – German	3,327,935	Latin	.de, .at, .ch, .li
5 ell – Greek	156,889	Greek	.gr, .cy
6 eng – English	25,100,666	Latin	.uk, .us, .au, .ca, …
7 fin – Finnish	140,139	Latin	.fi
8 fra – French	2,068,910	Latin	.fr, .be, .sn, …
9 heb – Hebrew	106,095	Hebrew	.il
10 hun – Hungarian	176,402	Latin	.hu
11 ind – Indonesian	423,909	Latin	.id
12 ita – Italian	868,195	Latin	.it, .sm, .va
13 jpn – Japanese	2,536,128	CJK	.jp
14 kor – Korean	289,573	Hangul	.kr, .kp
15 lat – Latin	815,813	Latin	(none)
16 nld – Dutch	978,614	Latin	.nl, .be
17 nor – Norwegian	130,172	Latin	.no
18 pol – Polish	599,296	Latin	.pl
19 por – Portuguese	751,436	Latin	.pt, .br, .ao, .mz, …
20 ron – Romanian	134,476	Latin	.ro, .md
21 rus – Russian	1,316,615	Cyrillic	.ru, .by, .su
22 slk – Slovak	131,507	Latin	.sk
23 spa – Spanish	1,600,880	Latin	.es, .mx, .ar, .co, …
24 srp – Serbian	149,086	Cyrillic	.rs, .me
25 swe – Swedish	301,628	Latin	.se
26 tha – Thai	151,102	Thai	.th
27 tur – Turkish	404,312	Latin	.tr
28 ukr – Ukrainian	127,040	Cyrillic	.ua
29 vie – Vietnamese	280,515	Latin	.vn
30 zho – Chinese	2,654,024	CJK	.cn, .tw, .hk, .mo

Table 13: The 30 selected languages (>100 K domain-language occurrences), sorted by ISO 639-3 code.

exclusively *Reference* (84.9 %), and `.jp` domains are overwhelmingly *Business* (64.7 %).

B Qualitative Tokenization: Additional Example

A second qualitative example with the German domain `biosa-vitalkonzepte.de` confirms the trends observed in the main text (Section 4.2, Table 15).

The domain is a compound of the brand name *Biosa* and the German word *Vitalkonzepte* ("vitality concepts"), itself composed of the stem *vital* and the noun *Konzepte* ("concepts"). Our unigram tokenizer recovers both meaningful components (`vital`, `konzepte`), whereas most general-purpose tokenizers fragment them into linguistically opaque pieces such as `alk on ze pte` or `kon ze pt e`.

C Tokenizer Normalization Details

To compare tokenizers fairly, we normalize each one to reflect only actual text content. Without this, metrics such as piece length and compression ratio would be corrupted by tokenizer-specific artifacts. We also remove special tokens such as start-of-sentence markers.

The following normalizations are applied where needed:

- **WordPiece ## stripping** (BERT, mBERT): WordPiece marks continuation subwords with a leading ## (e.g. `bio ##sa`). We strip it to avoid inflating piece lengths by two characters.

- **SentencePiece _ stripping** (Llama 2, ALBERT, BigBird, XLNet, T5): SentencePiece prepends the meta-character _ (U+2581) to mark word or input boundaries. Since domain names contain no spaces, only the first token is affected. We remove it so that character lengths and span positions remain correct.

- **XLNet first-token removal**: XLNet's HuggingFace tokenizer prepends an extra token when encoding without special tokens. We discard it to avoid inflating the token count.

- **No normalization needed**: Our domain-trained SentencePiece models never learned _ since training data contained no spaces. Byte-level BPE tokenizers (BLOOM, RoBERTa) encode spaces as Ġ, which never appears in domain names. Tiktoken-based models (GPT-4, Llama 3), Mistral Tekken, and DeepSeek return clean subwords without boundary prefixes.

Although some tokenizers above need no prefix stripping for domain names, their byte-level BPE vocabularies are still trained on whitespace-delimited text. A leading space changes the first token they emit. For example, when encoding `smokeheadguitars.co.uk` *without* a leading space, GPT-4 and Llama 3 begin with `sm`. Prepending a space changes the first token to the whitespace-prefixed piece `␣smoke`, merging four additional characters into the initial token. The same pattern holds for DeepSeek (`sm` → `smoke`), BLOOM (`sm` → `Ġsmoke`), RoBERTa (`sm` → `Ġsmoke`), and Mistral Tekken (`sm` → `␣smoke`). The remaining tokenizers produce the same first token regardless of a leading space.

D Model Architecture Details

BiLSTM. Following Toftrup et al. (2021), designed for short-text language identification. A learned embedding layer maps token indices to embedding vectors, followed by a single bidirectional LSTM layer with 768-dimensional hidden states. The final hidden states from both directions are

TLD	Count	%	Top 5 categories
.com	318,090	39.3	Business (33.6 %), Shopping (12.4 %), Society (10.0 %), Arts (9.4 %), Recreation (8.5 %)
.de	105,337	13.0	Business (27.0 %), Society (12.3 %), Health (11.4 %), Sports (10.7 %), Arts (9.5 %)
.org	75,469	9.3	Society (43.4 %), Health (9.5 %), Arts (9.4 %), Recreation (8.3 %), Science (7.7 %)
.jp	28,827	3.6	Business (64.7 %), Society (6.7 %), Arts (5.4 %), Shopping (4.2 %), Sports (3.8 %)
.ru	24,011	3.0	Business (39.1 %), Society (8.8 %), Arts (8.0 %), Recreation (7.6 %), Shopping (7.1 %)
.net	23,882	2.9	Business (19.7 %), Society (15.5 %), Arts (13.4 %), Computers (11.0 %), Recreation (9.3 %)
.uk	23,798	2.9	Business (24.3 %), Society (14.2 %), Sports (12.7 %), Recreation (10.5 %), Arts (10.3 %)
.nl	23,295	2.9	Business (26.6 %), Recreation (13.5 %), Society (11.4 %), Sports (9.9 %), Health (8.0 %)
.it	22,144	2.7	Business (52.5 %), Society (9.0 %), Arts (8.8 %), Recreation (6.0 %), Shopping (5.7 %)
.ch	13,407	1.7	Business (26.6 %), Sports (16.2 %), Arts (10.1 %), Health (10.1 %), Society (9.9 %)
.dk	13,160	1.6	Business (28.6 %), Recreation (14.8 %), Society (9.5 %), Shopping (9.3 %), Health (8.6 %)
.pl	12,210	1.5	Business (47.9 %), Shopping (10.6 %), Society (7.2 %), Arts (6.5 %), Science (6.2 %)
.cz	8,943	1.1	Business (27.5 %), Kids_and_Teens (11.4 %), Arts (10.0 %), Recreation (9.5 %), Shopping (7.7 %)
.au	8,676	1.1	Business (25.5 %), Recreation (16.5 %), Society (15.8 %), Sports (9.1 %), Shopping (7.7 %)
.fr	8,160	1.0	Business (25.5 %), Arts (11.1 %), Society (9.3 %), Science (9.0 %), Health (8.9 %)
.info	4,462	0.6	Society (19.0 %), Business (14.0 %), Arts (12.8 %), Recreation (11.8 %), Health (9.4 %)
.edu	2,736	0.3	Reference (84.9 %), Society (8.9 %), Health (4.1 %), Science (2.2 %), Arts (1.9 %)
Other	93,141	11.5	

Table 14: Per-TLD class profiles in the *Curlie* dataset for the top 15 TLDs by count plus `.info` and `.edu` as notable generic and purpose-specific TLDs (out of 490 unique TLDs observed), showing the five most frequent content categories.

concatenated and projected through a linear layer to class logits.

Encoder. An encoder-only Transformer (Vaswani et al., 2017) with bidirectional self-attention. A learned embedding layer, with learned absolute positional embeddings feeds into `nn.TransformerEncoder`. Masked mean pooling over non-padding positions (Reimers and Gurevych, 2019) produces the sequence representation, which is projected to class logits.

Decoder. A decoder-only (GPT-style) Transformer (Radford et al., 2018) with causal self-attention. It shares the same embedding, scaling, and positional embedding scheme as the Encoder but uses `nn.TransformerEncoder` with a causal mask so that each position attends only to preceding positions. The hidden state of the final token is projected to class logits.

Transformer. A full encoder–decoder Transformer (Vaswani et al., 2017). The encoder processes the input token sequence using bidirectional self-attention. The decoder receives a right-shifted copy prepended with a start-of-sequence token and applies causal masking. Both encoder and decoder share the same embedding layer with learned absolute positional embeddings. The hidden state of the final decoder token is projected to class logits.

General-purpose tokenizers	
GPT-4:	`b iosa -v it alk on ze pte .de`
+ ws:	`␣bios a -v it alk on ze pte .de`
DeepSeek:	`b iosa -v ital kon ze pte .de`
+ ws:	`␣bi osa -v ital kon ze pte .de`
Llama 3:	`b iosa -v it alk on ze pte .de`
+ ws:	`␣bios a -v it alk on ze pte .de`
Llama 2:	`b ios a - v it alk on ze p te . de`
+ ws:	`␣b ios a - v it alk on ze p te . de`
RoBERTa:	`b ios a - v it alk on ze pt e . de`
+ ws:	`␣bi osa - v it alk on ze pt e . de`
BLOOM:	`b iosa -v ital k onz ep te . de`
+ ws:	`␣b iosa -v ital k onz ep te . de`
Mistral Tekken:	`b iosa -v ital kon z ep te .de`
+ ws:	`␣b iosa -v ital kon z ep te .de`
BigBird:	`bi osa - vi talk onz ep te . de`
T5:	`bio s a - vita l konzept e . de`
ALBERT:	`bio sa - vi talk on ze p te . de`
XLNet:	`bio sa - vi talk on ze pt e . de`
BERT:	`bio sa - vital kon ze pt e . de`
mBERT:	`bio sa - vital kon ze pte . de`
:	
In-domain & baseline tokenizers	
char baseline:	`b i o s a - v i t a l k o n z e p t e . d e`
our unigram-1K:	`bio sa - vi tal ko nz ep te . de`
our unigram-5K:	`bio sa - vital kon ze p te . de`
our unigram-10K:	`bio sa - vital konzept e . de`
our unigram-30K:	`bio sa - vital konzepte . de`
our unigram-50K:	`bio sa - vital konzepte . de`
our unigram-100K:	`bio sa - vital konzepte . de`
our BPE-1K:	`bio sa - v ital k on z ep te . de`
our BPE-5K:	`bio sa - vital kon zep te . de`
our BPE-10K:	`bio sa - vital kon zep te . de`
our BPE-30K:	`bio sa - vital kon zepte . de`
our BPE-50K:	`bio sa - vital konzepte . de`
our BPE-100K:	`bio sa - vital konzepte . de`

Table 15: Tokenization of `biosa-vitalkonzepte.de` across all evaluated tokenizers. Rows marked + *ws* show the output when a leading whitespace is prepended (Section 4.1).