UNIVERSITY OF CAMBRIDGE

Language Model Tokenization: Segmentation and Compression

Suchir Salhan

Department of Computer Science & Technology, University of Cambridge, UK

L95 2025/26 24 November 2025

Tokenization: Why Care?

Tokenization is at the heart of a lot of LLM weirdness.

Tokenization: Why Care?

Tokenization is at the heart of much weirdness of LLMs. Do not brush it off.

- Why can't LLM spell words? **Tokenization**.
- Why can't LLM do super simple string processing tasks like reversing a string? **Tokenization**.
- Why is LLM worse at non-English languages (e.g. Japanese)? **Tokenization**.
- Why is LLM bad at simple arithmetic? **Tokenization**.
- Why did GPT-2 have more than necessary trouble coding in Python? **Tokenization**.
- Why did my LLM abruptly halt when it sees the string "<|endoftext|>"? **Tokenization**.
- What is this weird warning I get about a "trailing whitespace"? **Tokenization**.
- Why the LLM break if I ask it about "SolidGoldMagikarp"? **Tokenization**.
- Why should I prefer to use YAML over JSON with LLMs? **Tokenization**.
- Why is LLM not actually end-to-end language modeling? **Tokenization**.
- What is the real root of suffering? **Tokenization**.

(Source: Andrej Karpathy)

Tokenization: Why Care?

Three Perspectives:

1. ML Engineer
2. The Linguist
3. NLP Researcher, *balancing both extremes*

Perspective 1: ML Engineer

Why can't LLMs spell words? Why can't LLMs reverse a string? Why is Llama3 rude? Tokenizers, the one thing that can't be changed without retraining.

Perspective 2: The Linguist

- Two general views on **morphology**: decompositional or non-decompositional. Most Cambridge researchers assume the former (and are probably right...)
- What is Morphology? Syntax all the way down, a module in its own right ...

Perspective 3: The NLP Researcher

Even when more data is available, Linguistics can be of value: it can inform [...] ***more language-balanced tokenization*** *(Creutz and Lagus, 2002; Limisiewicz et al., 2024; Fusco et al., 2024). From* OPITZ ET AL. (2025)*, Natural Language Processing RELIES on Linguistics*

Subword Tokenization

- BPE, **WordPiece** and **UnigramLM**

Subword Tokenization: BPE

merge 1	p i c k e d	p i c k l e d	p i c k l e s
merge 2	pi c k e d	pi c k l e d	pi c k l e s
merge 3	pi ck e d	pi ck l e d	pi ck l e s
merge 4	pick e d	pick l e d	pick l e s
merge 5	pick ed	pick l ed	pick l e s
final	pick ed	pickl ed	pickl e s

Subword Tokenization: WordPiece

WordPiece is the tokenization algorithm Google developed to pretrain BERT. It's very similar to BPE in terms of the training, but the actual tokenization is done differently.

Subword Tokenization: UnigramLM

Unigram LM tokenization takes the vocabulary $\mathcal{V}$ and unigram LM parameters and performs Viterbi inference to decode the segmentation with maximum likelihood under θ .
In contrast to the bottom-up construction process of Byte Pair Encoding, it begins with a superset of the final vocabulary, pruning it to the desired size.

Subword Tokenization: UnigramLM

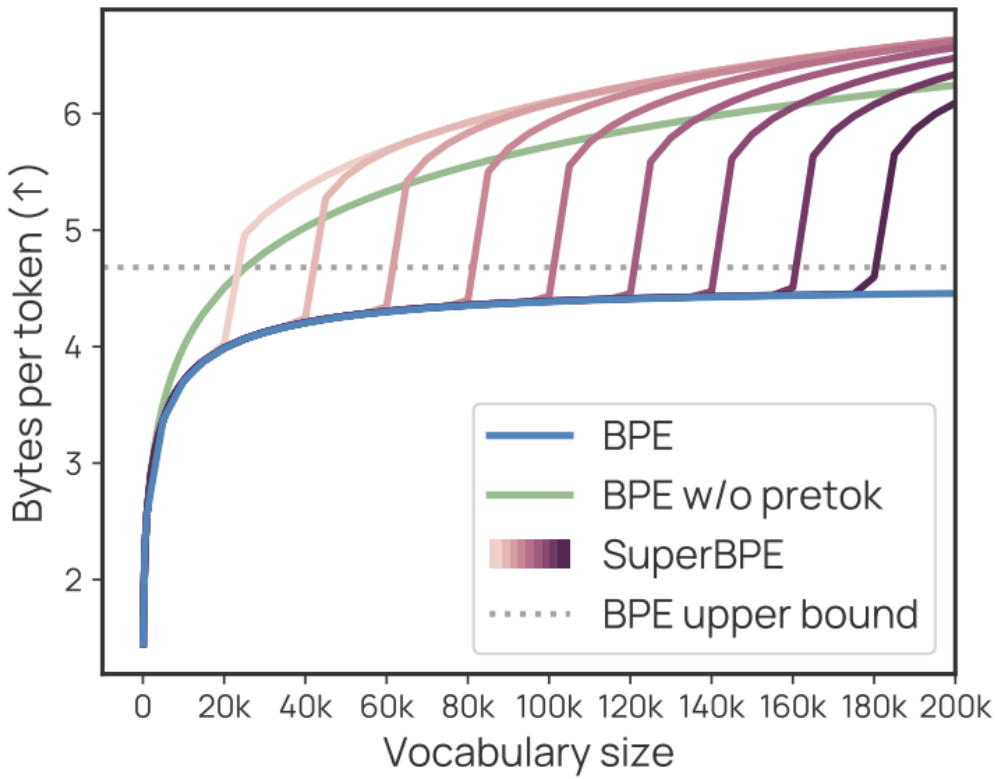
Algorithm 2 Unigram LM (Kudo, 2018)

Input: set of strings D, target vocab size k
procedure UNIGRAMLM(D, k)
    $V \leftarrow$ all substrings occurring more than
        once in D (not crossing words)
    while $|V| > k$ **do** ▷ Prune tokens
        Fit unigram LM θ to D
        for $t \in V$ **do** ▷ Estimate token 'loss'
            $L_t \leftarrow p_\theta(D) - p_{\theta'}(D)$
            where θ' is the LM without token t
        end for
        Remove $\min(|V| - k, \lfloor \alpha |V| \rfloor)$ of the
        tokens t with highest L_t from V,
        where $\alpha \in [0, 1]$ is a hyperparameter
    end while
    Fit final unigram LM θ to D
    return V, θ
end procedure

Pretokenization

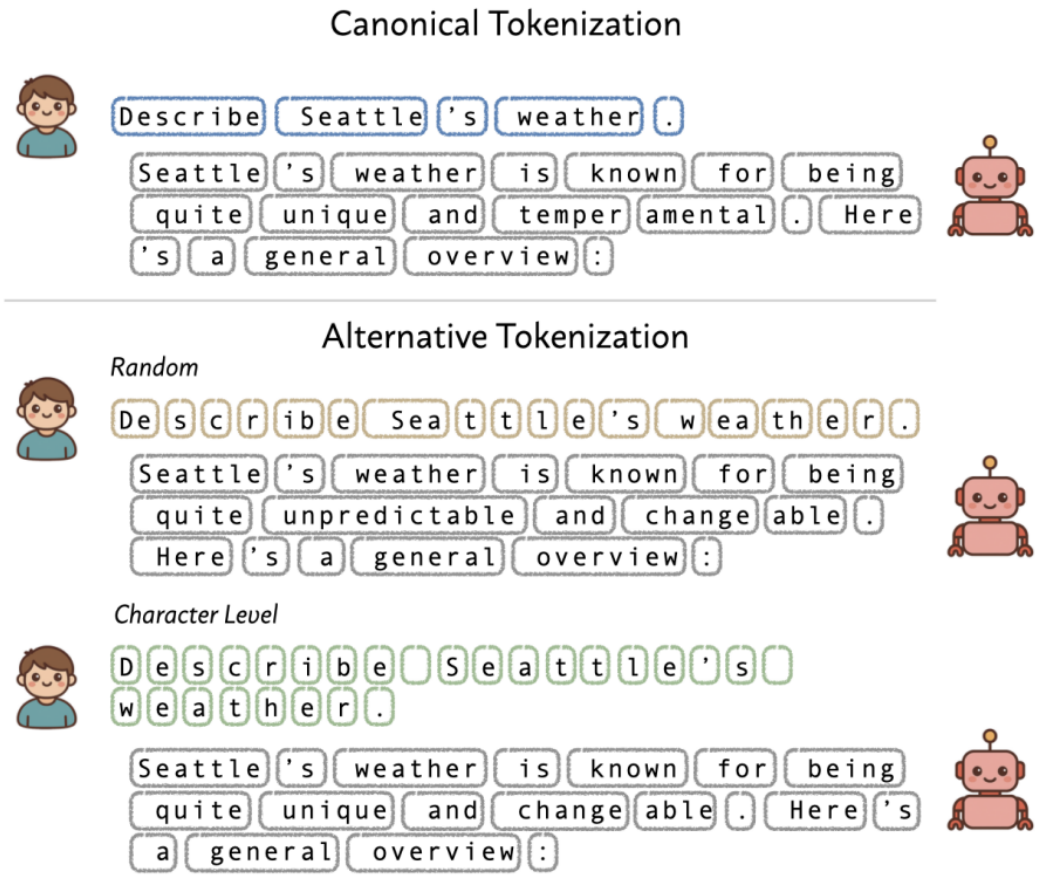
BPE: By the way , I am a fan of the Milky Way .

SuperBPE: By the way , I am a fan of the Milky Way .



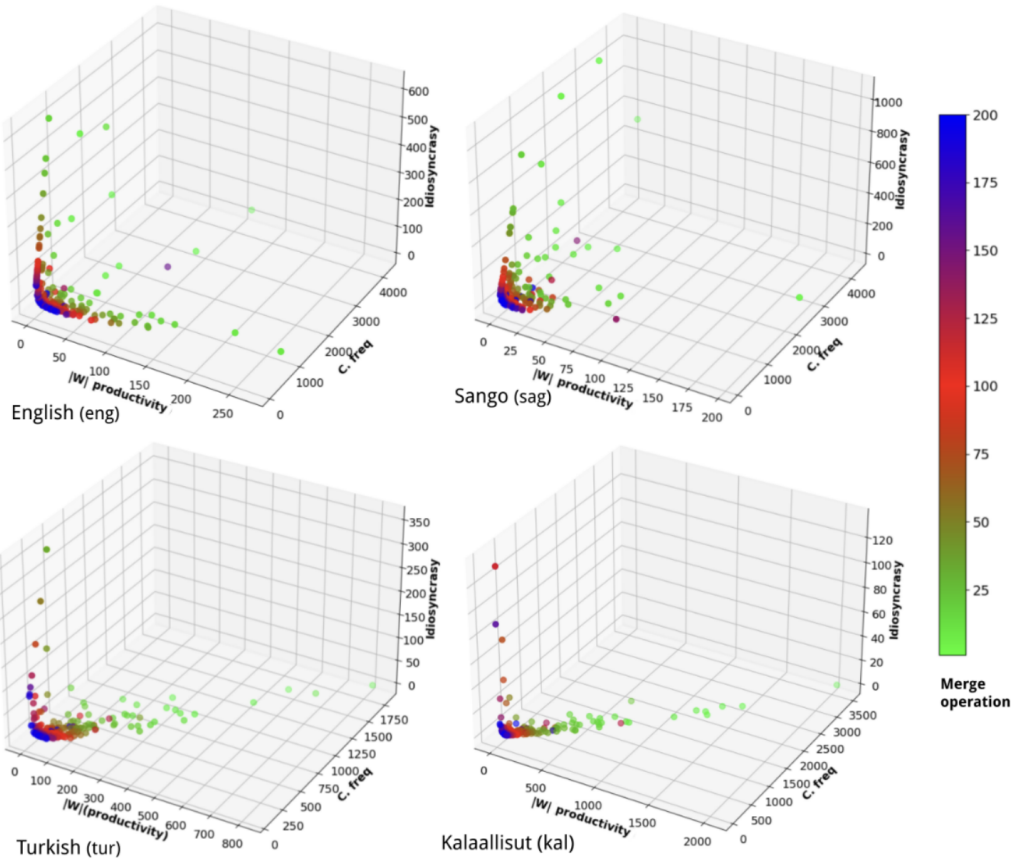
Non-Canonical Tokenizations

→ Cao & Rimmell (2021). (Figure from Zheng et al (2025, NEURIPS))



Through the Looking Glass of BPE Compression

Gutierrez-Vasques, Bentz & Samardžić (2023)



Intrinsic Evaluation

Table 1. Intrinsic evaluation results comparing BPE and BPE-WP to our ByteSpan tokenisers using surprisal across three vocabulary sizes. Scores are given to three significant figures, with the best score for each vocabulary size marked in **bold**.

Vocab Size	Tokenizer	Constraint	Learning Method	Morph. Alignment	Cognitive Plausibility	Fertility	Renyi Efficiency
16k	BPE	-	-	.694	**.302**	1.21	.468
	BPE-WP	-	-	.834	.297	**1.19**	.472
	ByteSpan	Global	Increment	**.899**	.146	1.90	.470
	ByteSpan	Monotonic	Frequency	.885	.254	1.39	**.483**
	ByteSpan	Monotonic	Seeding	.862	.272	1.22	.476
	ByteSpan	Combined	Frequency	.890	.268	1.29	.477
	ByteSpan	Combined	Seeding	.867	.279	1.21	.474

ByteSpan (Goriely, Salhan, Lesci, Cheng & Buttery 2025)

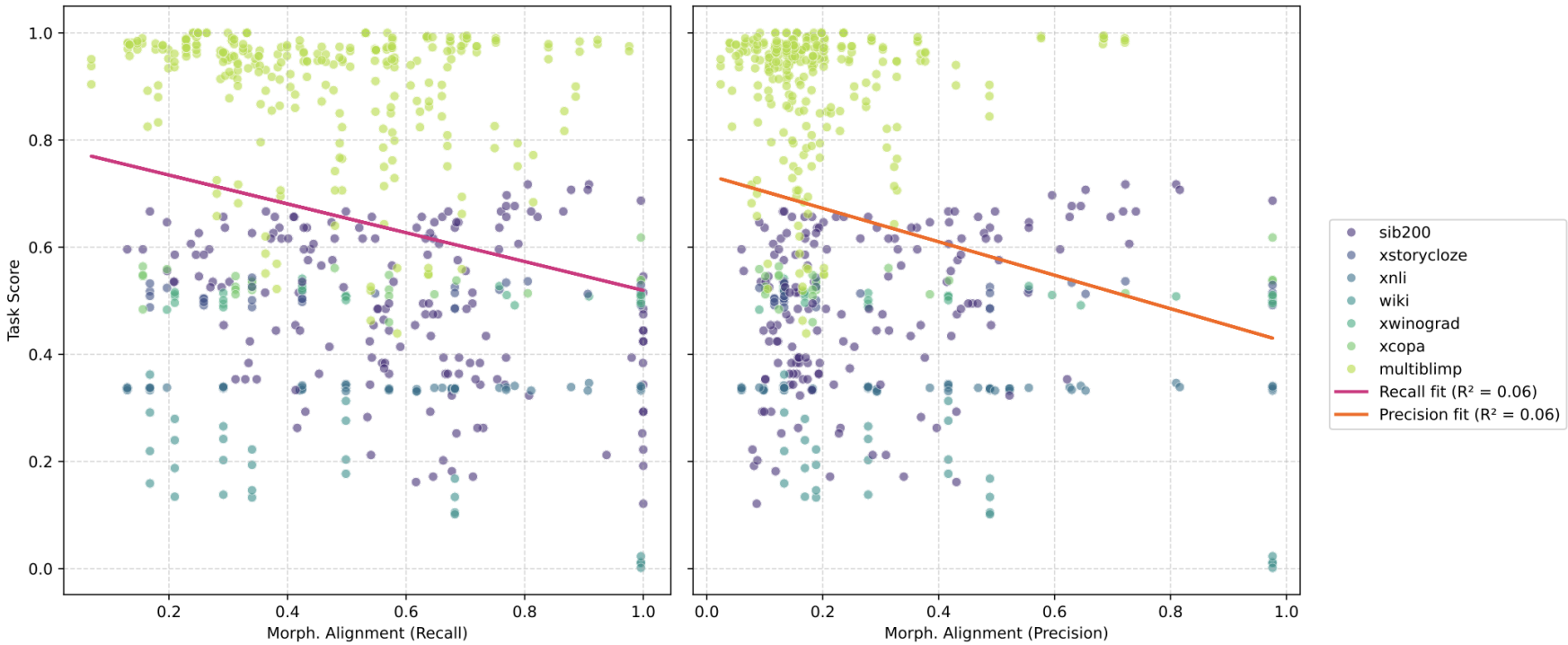
MorphScore

Table 1 illustrates how MorphScore is computed.

Language	**Word**	**Segmentation**	**Tokenizer Output / Score**
Basque	*aldiz*	aldi + z	[al, diz] $\rightarrow$ 0 [aldi, z] $\rightarrow$ 1
Croatian	*suučešnika*	suučešnik + a	various segmentations $\rightarrow$ 0
Icelandic	*samrás*	samrá+ s	[samra, s] $\rightarrow$ 1
Greek		+	[, , ,] $\rightarrow$ 1

Table: Example MorphScore calculations for several languages.

MorphScore Results



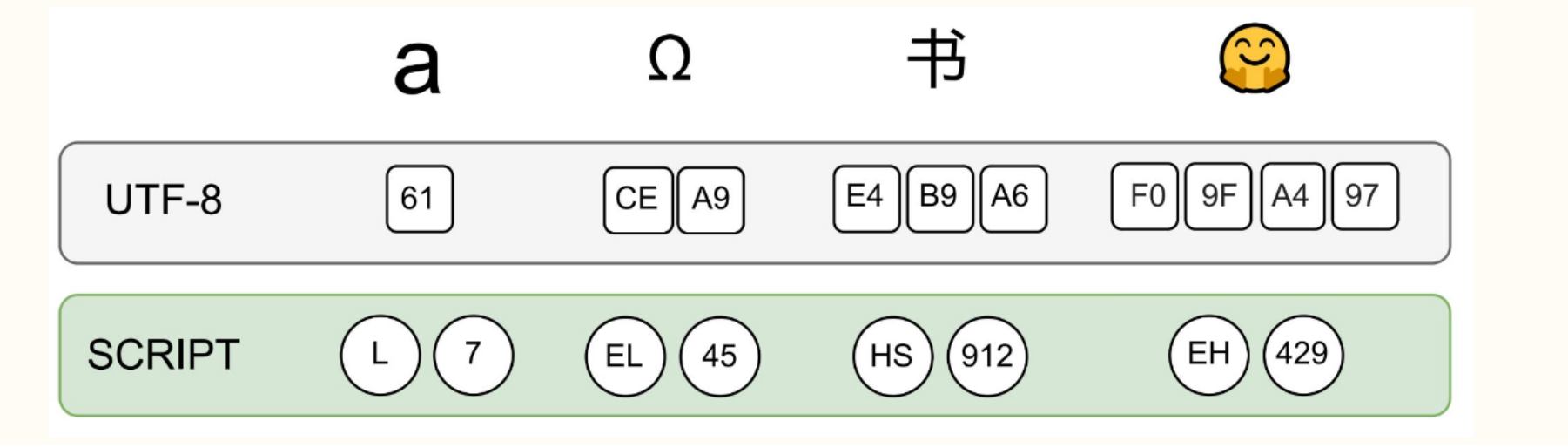
Arnett et al (2025)

Handling Script-Diversity

Characteristic	UTF-8 has	Useful for LLMs
Backward-compatible with ASCII	✅	❌
Robust to transmission errors	✅	❌
Semanticity	❌	✅
Multilingual Fairness	❌	✅

Slide from Sander Land (2025)

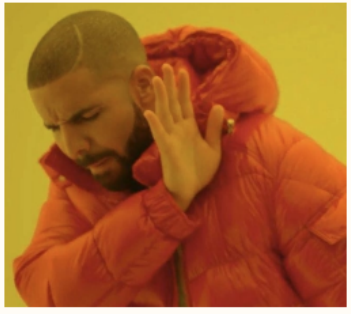
Handling Script-Diversity

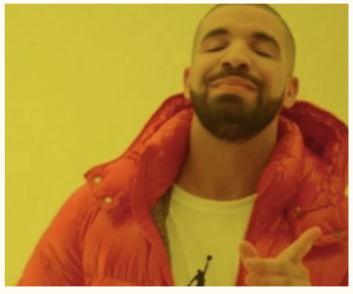


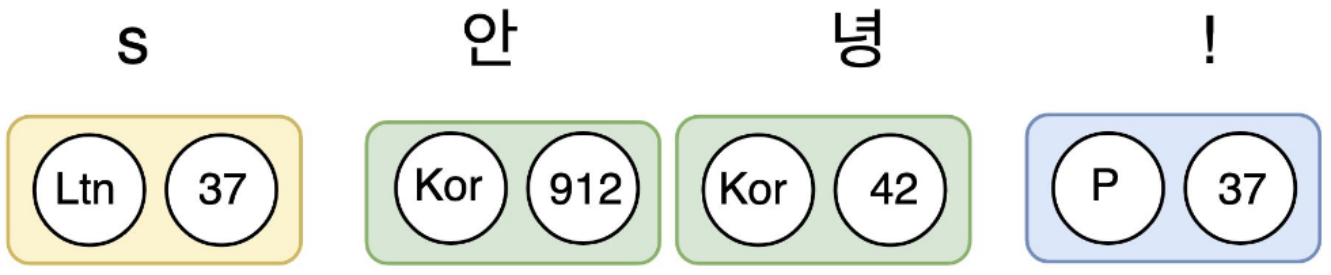
Slide from Sander Land (2025)

Handling Script-Diversity

```
[^\r\n\p{L}\p{N}]?[\p{Lu}\p{Lt}\p{Lm}\p{Lo}\p{M}]*
[\p{Ll}\p{Lm}\p{Lo}\p{M}]+(?i:'s|'t|'re|'ve|'m|'ll|'d)?|
[^\r\n\p{L}\p{N}]?[\p{Lu}\p{Lt}\p{Lm}\p{Lo}\p{M}]+
[\p{Ll}\p{Lm}\p{Lo}\p{M}]*(?i:'s|'t|'re|'ve|'m|'ll|'d)?|\p{N}
{1,3}| ?[^\s\p{L}\p{N}]+[\r\n/]*|\s*[\r\n]+|\s+(?!\S)|\s+
```







Slide from Sander Land (2025)

Tokenization Research in a Downstream World

Beat them

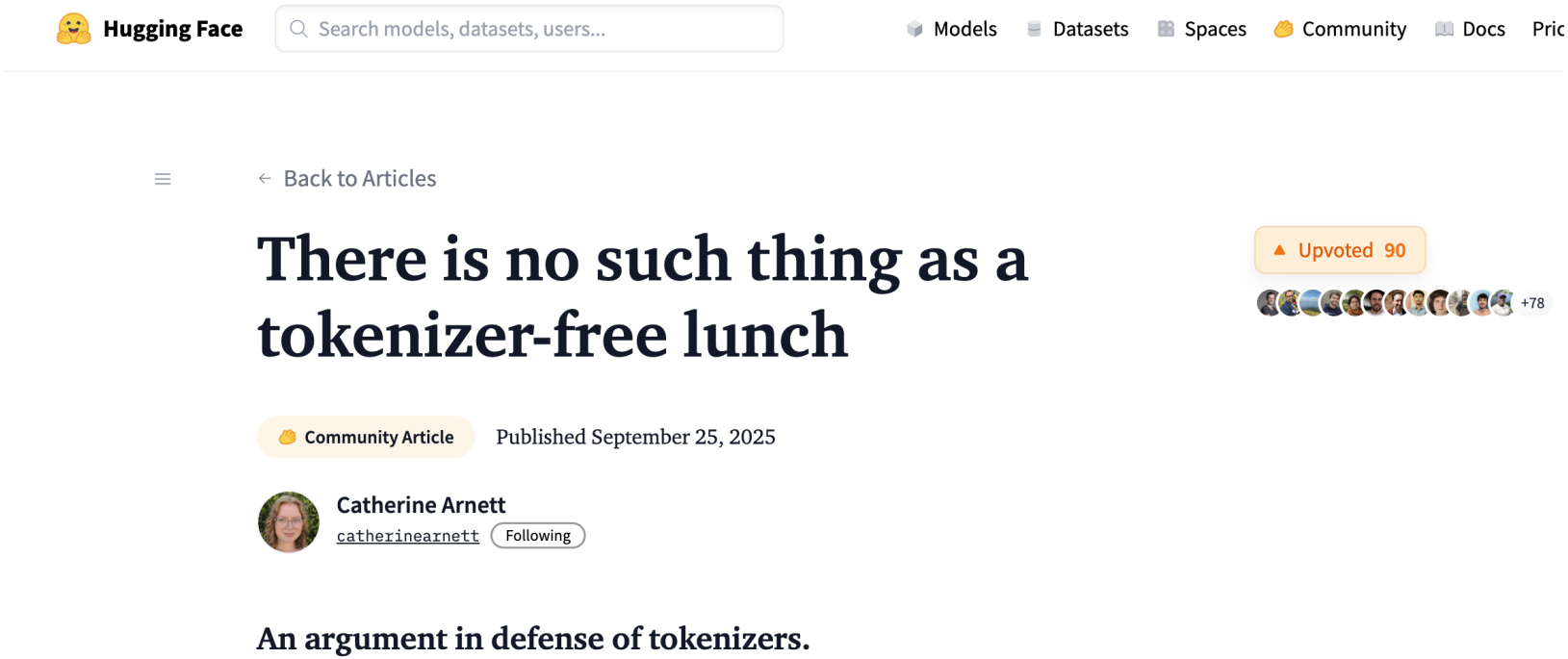
Creator: Virtisus | Credit: Getty Images

Join them

Fix them

Yuval Pinter's ICML TokShop Keynote

Tokenizer-Free?



Blogpost: In Defense of Tokenizers

Why Care about Tokenizers?

- Safety: E.g., **glitch tokens**. A while back people noticed some strange tokens in GPT-J (GPT-2) (“ RandomRedditorWithNo”, “davidjl”, “ SolidGoldMagikarp” and “BuyableInstoreAndOnline”)