LM4UC @ IJCAI 2026 • Bremen, Germany • 16 August 2026
Paper: arXiv:2608.09046
Code: github.com/HeyAvijitRoy/tea-benchmark

Measuring the Tokenization Premium

A Cost Audit for Underserved Language Communities | TEA Benchmark

Presented at: 35th International Joint Conference on Artificial Intelligence, 2026 Workshop
Language Models for Underserved Communities

Avijit Roy
John Jay College of Criminal Justice, CUNY
Proma Roy
The City College of New York, CUNY
Hrishitva Patel
University of Texas at San Antonio

1.56× Bengali GPT-4o token premium

≈4.5× Bengali under Qwen2.5 / Mistral

82k English-equivalent content in 128k window

2.37× Yoruba GPT-4o premium despite Latin script

Motivation

What does equivalent instructional content cost to represent before the model is invoked?

- Most multilingual evaluation starts after tokenization.
- TEA audits the hidden infrastructure layer before inference.
- This is one measurable form of structural silence.

Text → **Tokenizer** → Tokens → Model

One Item Example

"IndexError: list index out of range" · GPT-4o o200k_base

English	Input text:	IndexError: list index out of range	8 tokens
	Segments:	Index · Error · : · list · index · out · of · range	
Bengali	Input text:	IndexError: তালিকার সূচক সীমার বাইরে	11 tokens
	Segments:	Index · Error · : · তাল · কার · সূ · চ · ক · সীম · ার · বাইরে	
Hindi	Input text:	इंडेक्स त्रुटि: सूची इंडेक्स सीमा से बाहर	11 tokens
	Segments:	इ · ंडे · क्स · त्रु · टि · : · सूची · इंड · ेक्स · सीमा · बाहर	
Tamil	Input text:	குறியீட்டு பிழை: பட்டியலில் குறியீடு வரம்பிற்கு வெளிய உள்ளது	22 tokens
	Segments:	க · ுற · ிய · ீ · ட்டு · பு · ிழ · ை · : · பட்ட · ிய · லில் · குற · ிய · ீ · டு · வர · ம்ப · ிற்கு · வெளிய · ே · உள்ளது	
Yoruba	Input text:	Àṣìṣe Index: atọka kò sí ní ìwọn	19 tokens
	Segments:	À · ṣ · ì · ṣe · Index · : · à · k · ój · ọ · atọ · ka · kò · sí · ní · ì · wọ · ` · n	

English: 8 tokens, Tamil: 22 tokens. No model has run yet; the difference is produced by the encoder vocabulary.

Why The Premium Matters

- API billing increases with token count.
- Longer encodings consume context faster.
- Local deployment still pays sequence-length cost in memory and latency.

Tokenizer design changes what a system can carry, compute, and afford.

Supported in part by NSF ACCESS allocation CIS260616; analysis conducted on Indiana University Jetstream2.

Benchmark Design

A custom programming corpus

- 120 beginner Python debugging prompts.
- Three tiers: phrases, explanations, longer tutoring passages.
- Identifiers like TypeError, len(), NoneType, and IndexError stay in English.
- Bengali/Hindi human-reviewed; Arabic/Tamil/Yoruba exploratory.

Tokenizers audited: GPT-4o o200k_base · Qwen2.5-7B · Mistral-7B-v0.1

Metrics

TFR — Token Fertility Ration
`token (language) / tokens(English)`
Token premium relative to English.

ECW — Effective Context Window
`nominal window / TFR`
English-equivalent content in a fixed window.

Cost
`price × overhead × requests`
Illustrative API premium; ratios are stable.

Code-switching Sensitivity

Does retained English inflate the Bengali premium?

No. Clean Bengali is more expensive in every tier,
so retained English identifiers pull the measured premium down.

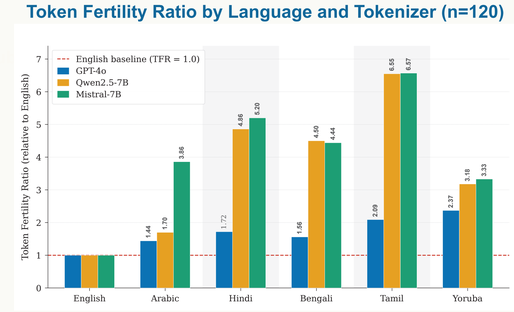
Subset	Tier 1	Tier 2	Tier 3
All Items	1.65	1.55	1.50
Clean Only	2.11	1.79	1.79

Clean = at least 75% Bengali-script alphabetic characters.

Therefore the 1.56× Bengali GPT-4o premium is conservative.

Results

Token Fertility Ratio by Language and Tokenizer (n=120)



Every tested non-English language is above English parity. Yoruba shows that Latin script is not enough to guarantee low token overhead.

Source: TEA Table 1. English baseline = 1.00.

Context and Data Impact

A nominal 128k window is not 128k of equivalent content for every language.

128k	82k	61k	54k
English: 100%	Bengali: 64%	Tamil: 48%	Yoruba: 42%

Dataset construction also scales:
100k English → 156k Bengali → 209k Tamil → 237k Yoruba

For tutoring systems, this means less room for prior turns, examples, retrieved documentation, and code.

Recommendations & Scope

1. Report tokenizer efficiency by language.
2. Publish context-equivalent capacity.
3. Audit terminology-heavy domains separately.
4. Measure sequence overhead before choosing deployment.

Scope: TEA measures infrastructure cost, not answer quality.
Bengali/Hindi are primary validated cases; Arabic/Tamil/Yoruba are exploratory.

Tokenization is a measurable equity layer beneath the model.
A vocabulary file and a parallel corpus are enough to expose representation overhead before inference.

Scan Me
Data & References

Interactive Poster and References: avijitroy.com/research/tea-benchmark