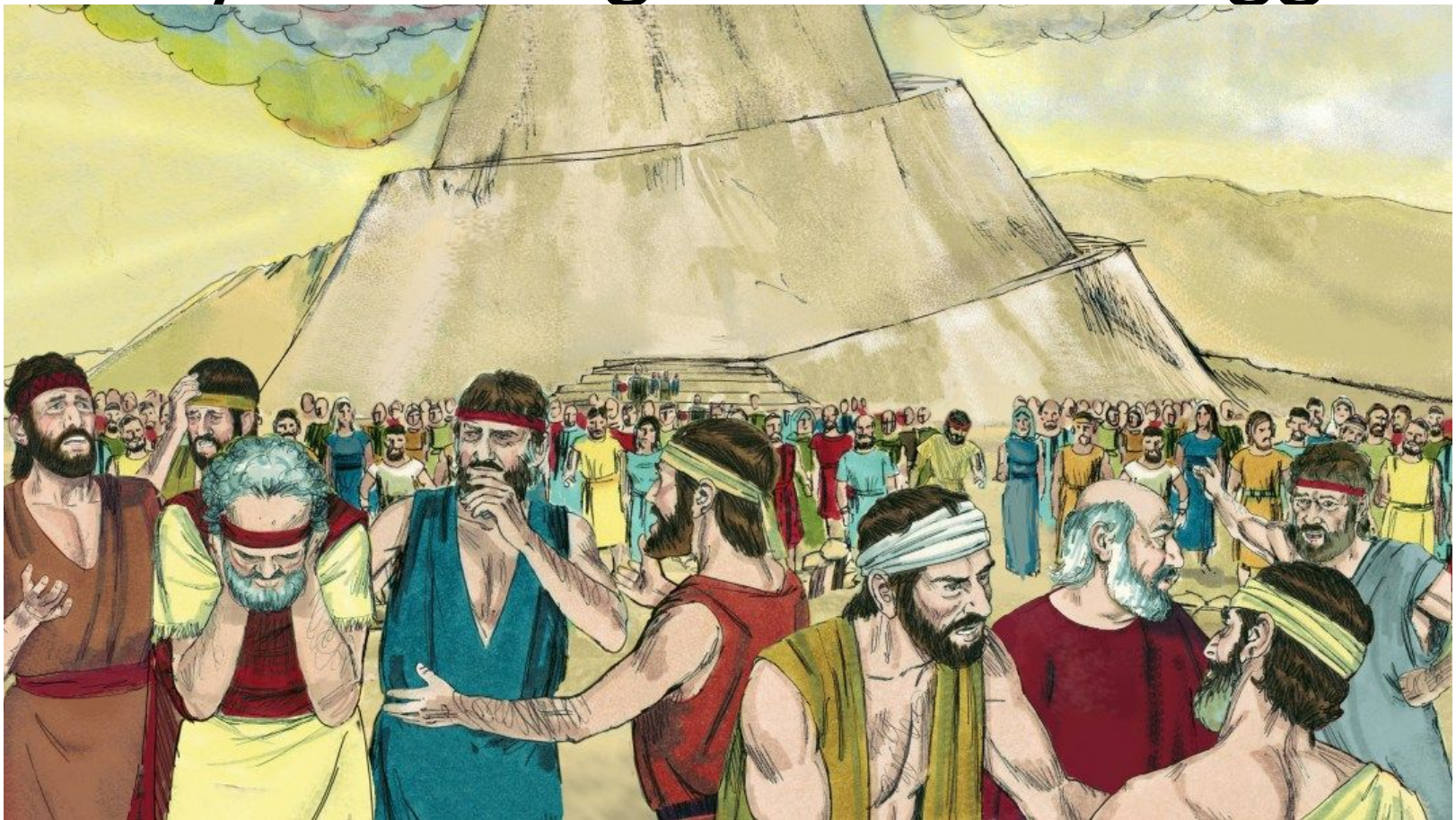
From English-Centric to Effective Bilingual: LLMs with Custom Tokenizers for Underrepresented Languages

Artur Kiulian, Anton Polishko, Mykola Khandoga, Yevhen Kostiuk,
Guillermo Gabrielli, Lukasz Gagala, Fadi Zaraket, Qusai Abu Obaida,
Hrishikesh Garud, Wendy Wing Yee Mak, Dmytro Chaplynskyi,
Selma Belhadj Amor, Grigol Peradze

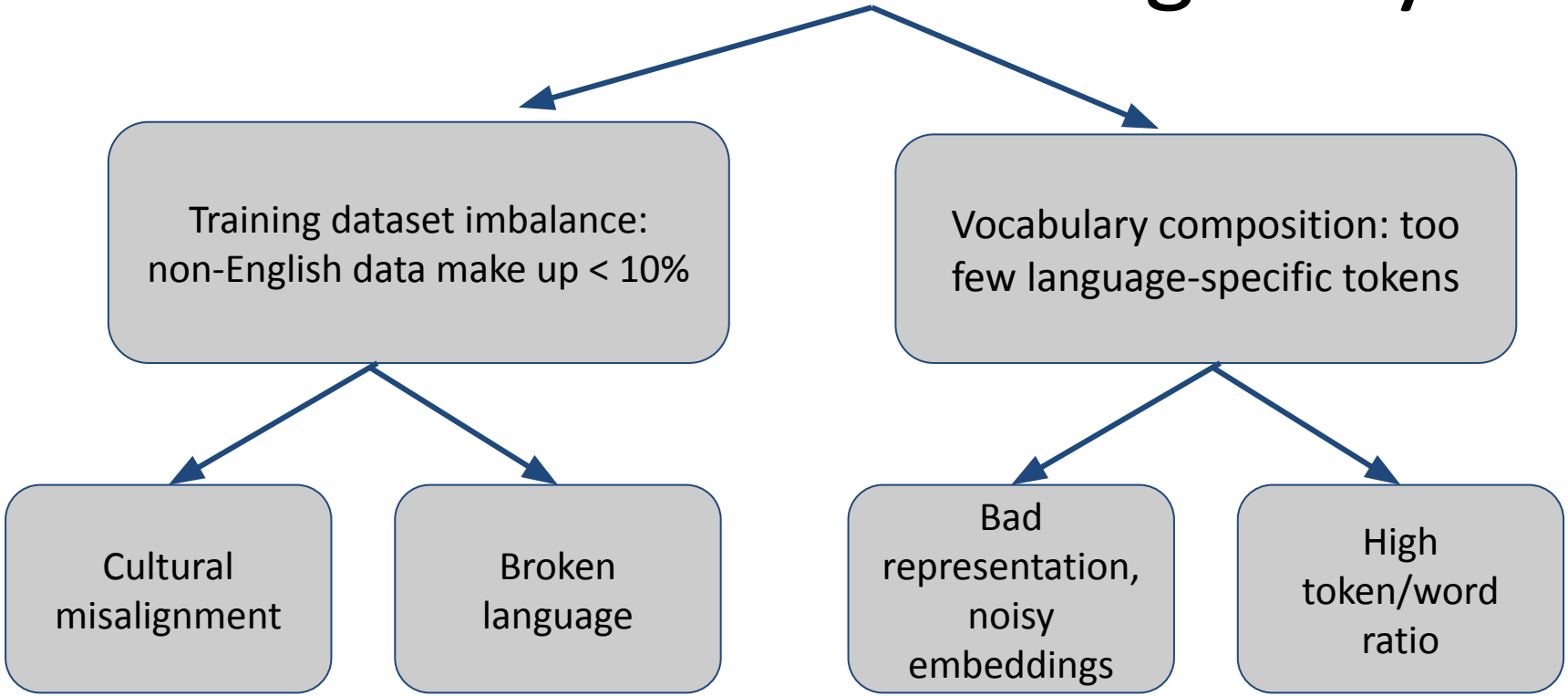
UNLP 2025

OpenBabylon

Why Multilingual LLMs Struggle



The curse of multilinguality

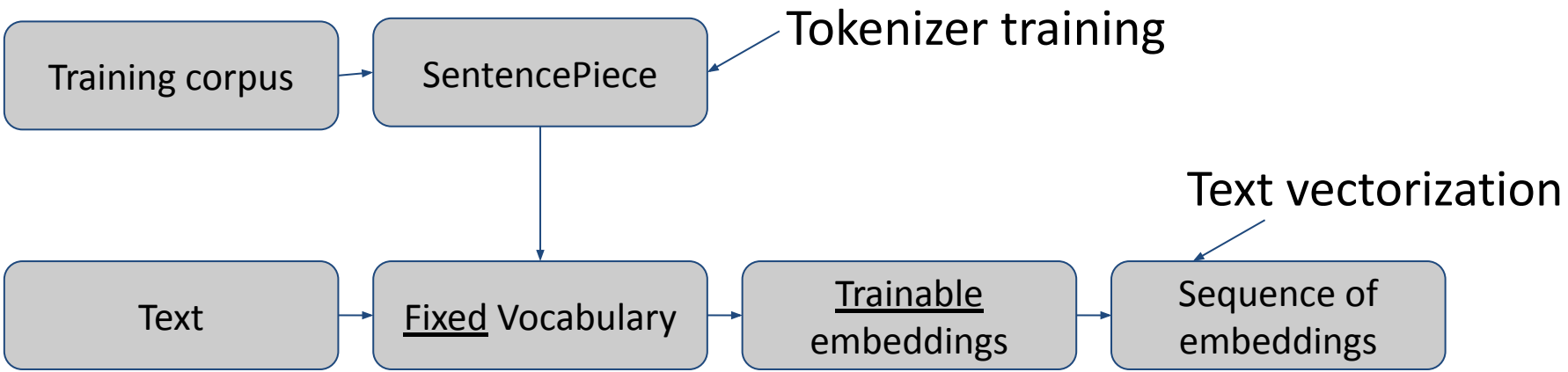


Our approach includes

1. Partial vocabulary replacement
2. Reinitalization of new token embeddings
3. Continual pretraining, ensuring the adoption of the new tokens
4. Eventual benchmarking

Tokenization 101

- We need tokenization to turn natural text into vector embeddings.
- Each LLM has a fixed vocabulary of sub-words, each sub-word has a trainable embedding vector, thus natural text is vectorized.
- Larger vocab allows shorter token sequences and more nuanced text representation.



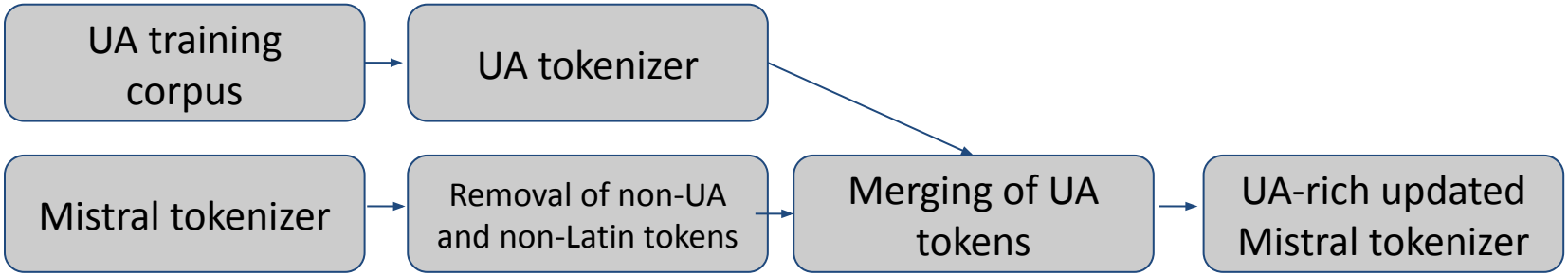
Mistral tokenization example

Input: The quick brown fox
Tokenizer → [The] [Ġquick] [Ġbrown] [Ġfox]
↑
well-formed English - low token/word fertility

Input: Швидкий рудий лис
Tokenizer → [Шв] [ид] [кий] [Ġ] [ру] [д] [ий] [Ġ] [лис]
↑
fragmented Cyrillic - high fertility

Input: الثعلب البني السريع
Tokenizer → [ال] [ث] [ع] [ل] [ب] [Ġ] [ا] [ل] [ب] [ن] [ي] [Ġ] [ا] [ل] [س] [ر] [ي] [ع]
↑
ultra-fragmented Arabic - very high fertility

Vocabulary update update workflow



This workflow allows to expand the target language vocab via cannibalizing non-Latin tokens.

1. Non-latin and non-UA tokens are removed along with their merge rules.
2. Token IDs of the removed tokens are assigned to additional UA tokens (existing UA tokens preserve their IDs)
3. Merge rules are updated to accommodate new UA tokens
4. Time to update embeddings

Tokenization performance

Mistral	Vanilla		*Ours*	
	Tokens	**Fertility**	**Tokens**	**Fertility**
Ukrainian	1,077	3.35	5,552	2.55
Arabic*	70	3.3	3,618	1.68
Georgian	29	7.61	5,531	2.68

Gemma	Vanilla		*Ours*	
	Tokens	**Fertility**	**Tokens**	**Fertility**
Ukrainian	6,426	2.55	75,704	1.56
Arabic*	6,075	1.65	32,333	1.52

*tokens/words ~ -log(vocab size)

Embeddings reinint

While the embeddings are trainable, we've given new tokens a warm start using the following heuristics:

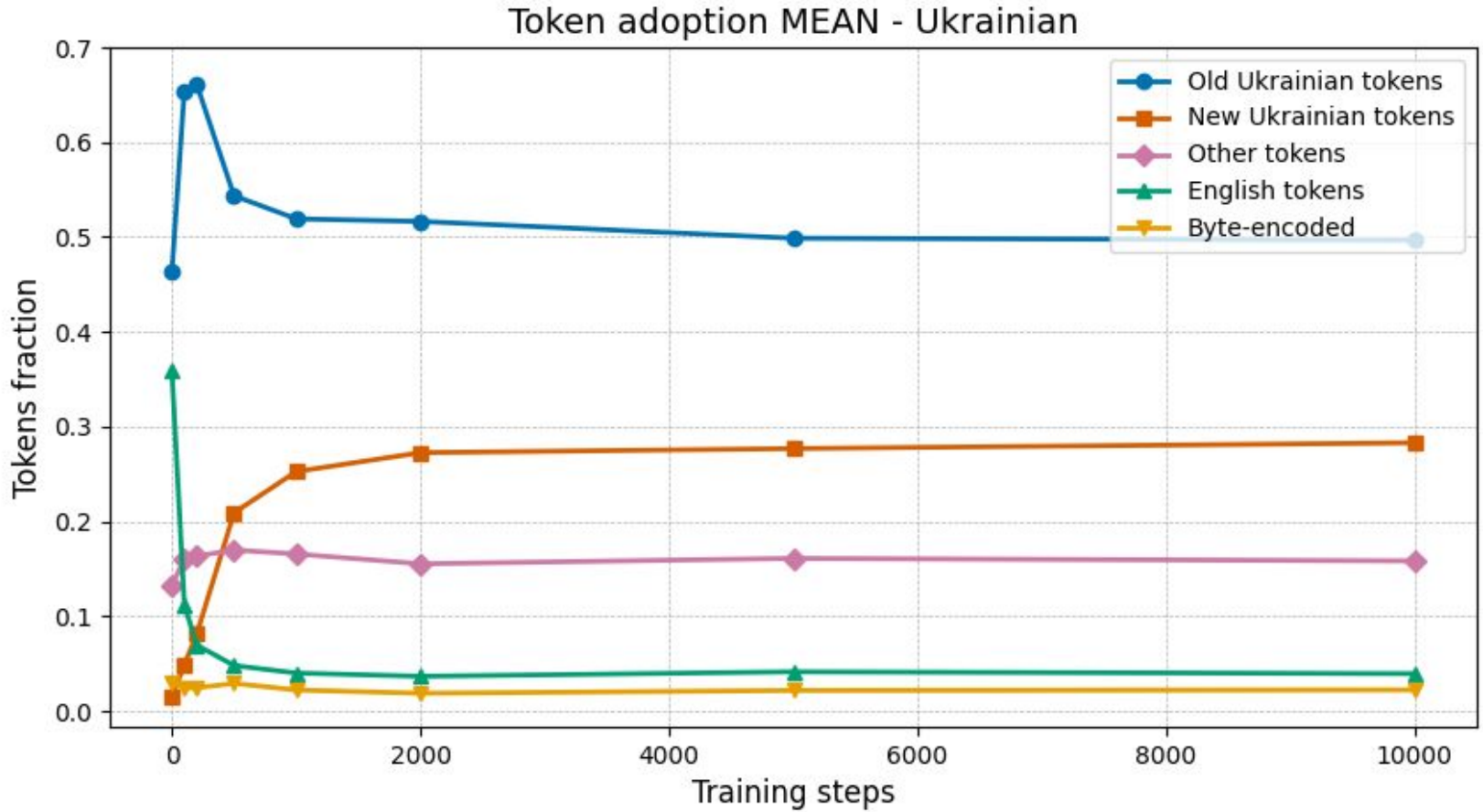
$$E(t_{new}) = \frac{1}{n} \cdot \sum_i E(t_i)$$

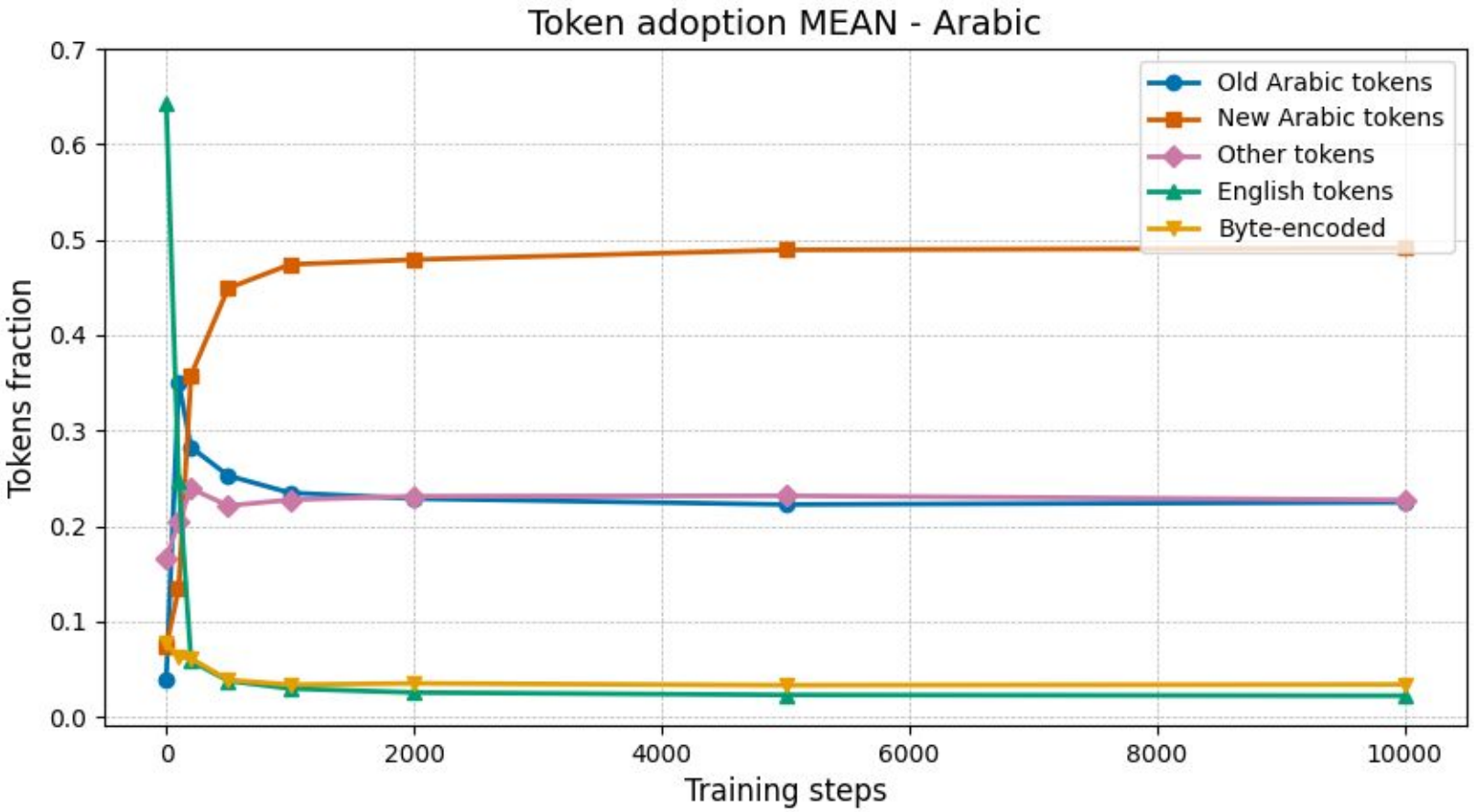
where the new (longer) tokens are expressed through existing tokens and their respective embedding vectors are initialized as their normalized linear sum.

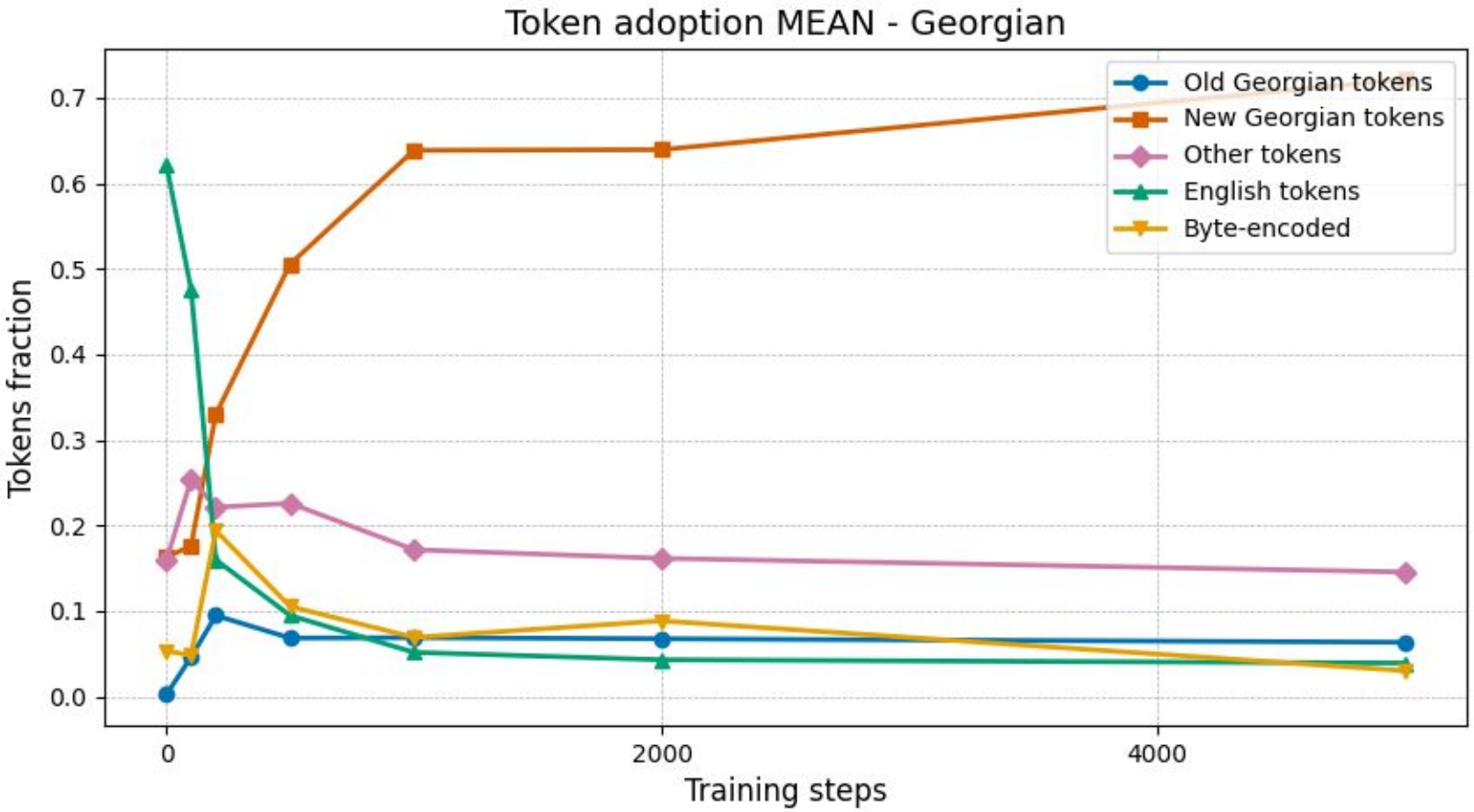
Continual pre-training

We have used 8xA100-80-GB for the continual pre-training of our models.

The following graphs illustrate the adoption of the new tokens during the continual pre-training over the first ~350M tokens (172M for Georgian).


Token adoption MEAN - Ukrainian
Old Ukrainian tokens
New Ukrainian tokens
Other tokens
English tokens
Byte-encoded
Tokens fraction
Training steps
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
2000
4000
6000
8000
10000


Token adoption MEAN - Arabic
Old Arabic tokens
New Arabic tokens
Other tokens
English tokens
Byte-encoded
Tokens fraction
Training steps
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
2000
4000
6000
8000
10000


Token adoption MEAN - Georgian
Old Georgian tokens
New Georgian tokens
Other tokens
English tokens
Byte-encoded
Tokens fraction
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
2000
4000
Training steps

Evaluation metrics

NEWR (Non-Existing Word Ratio): The percentage of generated words not present in a reference vocabulary — high NEWR indicates hallucinations or poor lexical quality.

CSWR (Code-Switching Word Ratio): Measures how often the model generates words mixing characters from different alphabets (e.g., Latin + Cyrillic) — a sign of garbled outputs.

GCS (Grammar Correctness Score): A normalized score (via GPT-4 or equivalent) assessing grammatical quality of generations — higher is better.

Evaluation results

Model	GCS ↑	NEWR ↓	CSWR ↓
Ukrainian			
Vanilla	0.264	0.089	0.515
Tuned	0.388	0.032	0.002
Ours	**0.503**	**0.030**	**0.001**
Arabic			
Vanilla	0.040	0.863	0.450
Tuned	0.238	0.079	0.004
Ours	**0.548**	**0.050**	**0.002**

Conclusions

- **Vocabulary is destiny:** Token size & composition are the hidden levers of quality—fix them and code-switching, hallucinated words & grammar errors go away.

- **Small tweak, big win:** A simple vocabulary-extension step lifted Ukrainian grammar +30 pts and Arabic +50 pts—far beyond what extra data alone achieved.

- **Equity in NLP:** Custom tokenizers open the door for low-resource scripts, cutting compute cost and making non-English LLMs economically viable.