Switch-Tokenizer: Pretraining Language Models to Use Multiple Tokenizers

Nikita Razuvaev

Data Scientist, MTS Fintech

GitHub: hardesttype/switch-tokenizer

April 20, 2025

SUMMER OF MACHINE LEARNING AT SKOLTECH

Introduction

- **What is Switch-Tokenizer?**
  A multilingual tokenizer implementation that uses a shared vocabulary space between different language-specific tokenizers.
- **Why is it important?**
  Enables efficient parameter usage in multilingual language models through context-dependent token interpretation.
- **Background**
  Traditional multilingual models use a common vocabulary trained on multilingual data, which can be very unbalanced, resulting in inefficient parameter usage and increased model size.
- **Goal of the research**
  Develop an efficient multilingual tokenization approach that maintains performance while reducing parameter costs.

Problem Statement

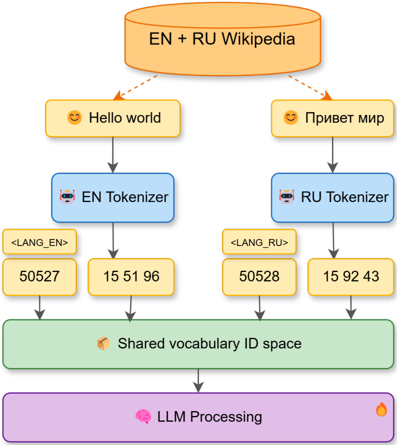
- **What exactly are we solving?**
  Inefficient parameter usage in multilingual language models due to common vocabularies trained on unbalanced multilingual data.
- **Challenges**
  - Maintaining a fixed-size embedding table despite multiple languages
  - Learning context-dependent token interpretation
  - Ensuring tokenization efficiency without using a single shared vocabulary
- **Scope**
  Focusing on efficient multilingual language modeling while maintaining performance across languages.

Related Work: Tokenizer Adaptation Methods

Method	Approach	Key Advantages
Zero-Shot Tokenizer Transfer	Transfers pretrained model to new tokenizer without fine-tuning	Enables switching tokenizers post-training with minimal performance loss
LazyLLM	Dynamic token pruning during inference	Reduces computation for long contexts by 2-4x while preserving quality
ReTok	Replaces original tokenizer with more efficient one	Improves context length by up to 2x with minimal perplexity degradation
MRT5	Dynamic token merging for byte-level models	Processes longer contexts efficiently while maintaining byte-level precision
Dynamic Tokenization	Retrofits LLMs with specialized tokenizers	Enables specialized encoding for different domains/languages without retraining

Methods: The Switch-Tokenizer Approach

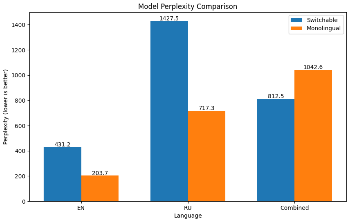
- **Approach:**
  - Each language has its own tokenizer with its own vocabulary
  - All tokenizers map into the **same shared vocabulary ID space**
- **Why this method?**
  Maintains a fixed-size embedding table and output projection layer regardless of the number of languages.
- **How it works:**
  The model learns to associate token IDs with different tokens depending on the language context.



Switch-Tokenizer methodology

Results: Experiment 1

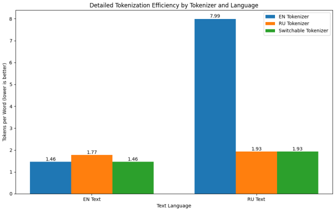
- **Key findings:**
  - With equal (monolingual) training budget for all models, monolingual models perform better on their respective languages
  - But for multilingual tasks, the switchable model outperforms by 22.07%
  - Tokenization efficiency remained consistent across approaches



Perplexity comparison (lower is better)

Tokenization Efficiency

- **Metrics used:**
  - Tokens per word ratio (lower is better)
  - Perplexity scores across languages
- **Experimental setup:**
  - Data: Wikipedia articles (EN + RU)
  - Base model: gpt2-medium
  - Tokenizers: gpt2 (EN), ruGPT-3.5-13B (RU)
- **Idea:**
  - Increase token budget to multilingual



Tokens per word comparison

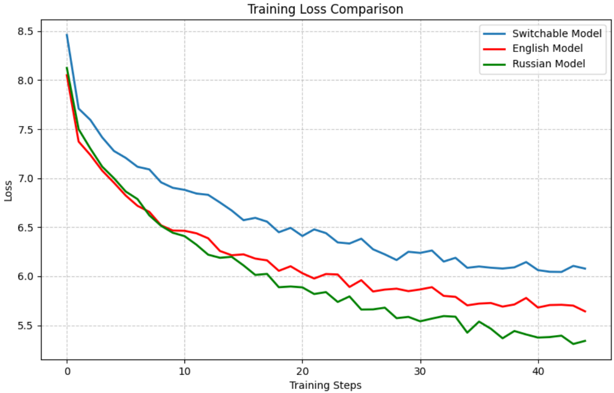
Future Work

- **Planned experiments:**
  - Comparison vs. Common Vocabulary Approach
  - Multilingual Baseline Comparison
  - Context Sensitivity Analysis
- **Unresolved challenges:**
  - Dynamic tokenizer switching without explicit language tokens
  - Scaling to larger models and more languages
- **Why it matters:**
  Efficient multilingual models have applications in translation, cross-lingual understanding, and content creation.
- **Future opportunities:**
  - Specialized tokenizers for programming languages
  - Expanded benchmarks on standard multilingual tasks

Bibliography I

- "Zero-Shot Tokenizer Transfer" (Minixhofer et al., 2024)
- "LazyLLM: Dynamic Token Pruning for Efficient Long Context LLM Inference" (Fu et al., 2024)
- "ReTok: Replacing Tokenizer to Enhance Representation Efficiency in Large Language Model" (Gu et al., 2024)
- "MrT5: Dynamic Token Merging for Efficient Byte-level Language Models" (Kallini et al., 2024)
- "Retrofitting Large Language Models with Dynamic Tokenization" (Feher et al., 2024)
- "Language Models are Unsupervised Multitask Learners" (Radford et al., 2019)
- "A Family of Pretrained Transformer Language Models for Russian" (Zmitrovich et al., 2023)
- "Wikimedia Downloads" (Wikimedia Foundation)
- "How does a Language-Specific Tokenizer affect LLMs?" (Seo et al., 2024)
- "Qtok: A Comprehensive Framework for Evaluating Multilingual Tokenizer Quality in Large Language Models" (Chelombitko et al., 2023)
- "Getting the most out of your tokenizer for pre-training and domain adaptation" (Dagan et al., 2023)
- "Tokenizer Choice For LLM Training: Negligible or Crucial?" (Ali et al., 2024)

Appendix: Training Curves



Training loss comparison between switchable and monolingual models