COST-EFFICIENT METHOD FOR LLM ADAPTATION FOR THE RUSSIAN LANGUAGE

Abstract. Adapting a pre-trained large language model (LLM) to a new language is a complex process requiring both expertise and substantial computational resources. While previous researches have focused on individual adaptation methods, a systematic evaluation remains lacking. We introduce a methodology for adapting a decoder-based LLM (Llama 3 Instruct) to Russian, utilizing the Token Substitution (TS) approach for modifying tokenizers. A detailed comparison is conducted, examining LLMs pre-trained on Russian, proprietary models. The evaluation considers text quality and token efficiency, using subsets of the SberQuAD dataset alongside a unified benchmark incorporating oasst2, ru_alpaca, and SynEL samples. Our findings underscore tokenizer adaptation as a cost-effective and practical strategy for improving LLM performance.

§1. Introduction

Adapting a pre-trained large language model (LLM) to a specific language can enhance its performance in that language. Previous studies have demonstrated this for both single-language [?, ?] and multi-language adaptation [?, ?]. In the majority of methods, modifying the vocabulary or tokenizer of a transformer-based [?] LLM serves as a crucial initial step, involving vocabulary adjustments followed by tuning of the modelвЂ™s weights and embeddings, fine-tuning (FT), and instruction tuning (IFT). The primary objective of adaptation is to shorten the input sequence length while improving quality by enabling the model to grasp the nuances, vocabulary, and grammar unique to the target languageвЂ™s linguistic context.

Token substitution is a tokenizer adaptation technique that modifies an existing vocabulary without expanding its size. This approach involves replacing infrequently used tokens with new ones and randomly initializing their embeddings [?]. However, existing adaptation methods are evaluated on different datasets, applied to various LLMs, and used across multiple languages, making direct comparisons inconsistent. As a result, tokenizer

Key words and phrases: key and words.

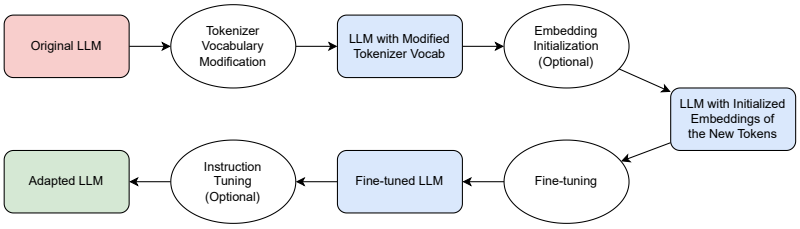

Fig. 1. General pipeline of LLM adaptation. The rounded colored rectangles represent a version of the LLM within the pipeline. The red rectangle is the input version of the LLM. The green rectangle is the final version of the LLM produced by the pipeline. The ellipses represent the stages of the pipeline.

adaptation remains largely a trial-and-error process. Additionally, it is unclear whether adapted models can outperform traditional continuous pre-training methods, which are often inefficient and require up to three times more computational resources.

This work aims to conduct a thorough evaluation of the tokenizer adaptation method using diverse metrics and benchmarks for a fixed language. We argue that standard LLM evaluation benchmarks are insufficient for assessing adapted models and introduce a specialized benchmark for this purpose. To evaluate LLM adaptation, we utilize three categories of metrics, demonstrating that competitive performance can be achieved without the need for resource-intensive continuous pre-training. Our primary contribution is the development and assessment of models that effectively balance performance and computational efficiency in LLM adaptation.

§2. Related Work

We can structure the process of adaptation as a pipeline consisting of multiple stages. Typically, this includes two essential stepsвЂ”vocabulary modification and fine-tuningвЂ”along with two optional steps: embedding initialization and instruction tuning. Figure 1 illustrates this pipeline.

Vocabulary modification involves adding new tokens to the modelвЂ™s tokenizer vocabulary. However, since these tokens initially lack input and output embeddings, it is often beneficial to initialize them using existing

embeddings. Fine-tuning is then performed to enhance the modelвЂ™s ability to process the target language effectively. An optional final step, instruction tuning, allows the model to better follow instructions in the adapted language.

In this work, we explore an approach to adapting an LLM tokenizer to the Russian language [?]. To ensure consistency and reproducibility, we conduct most experiments using the same base model, Llama-3-8B-Instruct, which is widely utilized in various adaptation efforts, including industry applications such as T-lite[1].

2.1. Token Substitution (TS). An alternative adaptation method has been proposed by [?]. This method maintains the original vocabulary size. Since the authors did not assign a specific name to this approach, we refer to it as the "Token Substitution" method. This technique involves replacing infrequently used tokens in the tokenizer with new ones. Initially, the method did not reuse existing token embeddings; instead, it assigned random values to the embeddings of newly introduced tokens. As a result, the approach depends on model pre-training to refine and optimize the embedding values.

§3. Dataset

Adaption techniques require training of the modified model to obtain the best token embeddings. We have gathered a single dataset of around 1 billion tokens from multiple different datasets in order to train the model. While the remaining text is in Russian, English makes up 25% of the text data. Because training on the Russian-only text could result in catastrophic forgetting, the English text was added to the dataset [?, ?, ?, ?, ?, ?, ?]. The dataset was cleaned and de-duplicated.

If you were collecting the dataset on your own please describe the collection procedure, like criteria were used to filter the documents, the preprocessing steps, etc. It is preferable that you release your dataset for the public, but you are not obliged to do this. Please make sure that you have legal rights to collect and distribute the data you were working with.

(F. Name) Address
E-mail: email

[1]https://huggingface.co/AnatoliiPotapov/T-lite-instruct-0.1