Prune or Retrain: Optimizing the Vocabulary of Multilingual Models for Estonian

Aleksei Dorkin and **Taido Purason** and **Kairit Sirts**
Institute of Computer Science
University of Tartu
{aleksei.dorkin, taido.purason, kairit.sirts}@ut.ee

Abstract

Adapting multilingual language models to specific languages can enhance both their efficiency and performance. In this study, we explore how modifying the vocabulary of a multilingual encoder model to better suit the Estonian language affects its downstream performance on the Named Entity Recognition (NER) task. The motivations for adjusting the vocabulary are twofold: practical benefits affecting the computational cost, such as reducing the input sequence length and the model size, and performance enhancements by tailoring the vocabulary to the particular language. We evaluate the effectiveness of two vocabulary adaptation approaches—retraining the tokenizer and pruning unused tokens—and assess their impact on the model's performance, particularly after continual training. While retraining the tokenizer degraded the performance of the NER task, suggesting that longer embedding tuning might be needed, we observed no negative effects on pruning.

1 Introduction

Adapting multilingual pretrained language models to specific languages can enhance both their efficiency and performance (Kuratov and Arkhipov, 2019; Mroczkowski et al., 2021). The adaptation generally involves continuously training the full model on language-specific data. This approach can be expensive and unsuitable for less-represented languages.

In this study, we explore how modifying the vocabulary of a multilingual language model to better suit the Estonian language affects its downstream performance. Compared to the previous works (Gee et al., 2022; Csaki et al., 2024; Tejaswi et al., 2024), we focus on training newly initialized embeddings rather than the specific initialization approaches. The motivations for adjusting the vocabulary are twofold:

1) Practical Benefits: A language-specific vocabulary can reduce the length of tokenized sequences, leading to more efficient training and fine-tuning. Meanwhile, the vocabulary of a multilingual model has to accommodate for all the languages it supports, which results in a significant amount of unused tokens in a monolingual use case. Consequently, adapting the vocabulary to a single language either by pruning the tokenizer or training a new one leads to the decrease in vocabulary size. In turn, decreasing the vocabulary size reduces the overall model size, which can improve computational efficiency.

2) Performance Enhancement: Tailoring the vocabulary to a particular language may improve the model's ability to understand and process text in that language, potentially boosting performance on language-specific tasks.

Our goal is to evaluate the effectiveness of two vocabulary adaptation approaches—retraining the tokenizer and pruning unused tokens—and assess their impact on encoder models' performance in the Estonian language, particularly after continual training, which is evaluated by fine-tuning on the named entity recognition task. In the retraining approach, we train a new tokenizer on the Estonian National corpus (ENC),[1] and use the resulting vocabulary to replace/adapt the encoder-based multilingual DeBERTa model (He et al., 2023). We first train the new embeddings with other model parameters frozen, and then continue training the full model with the masked language modeling loss. Finally, we fine-tune the model with new/adapted vocabulary on the Estonian NER dataset (Sirts, 2023) to evaluate the effect of vocabulary optimization. In the second approach, we simply prune the model's initial vocabulary to only keep the tokens that are

[1] https://doi.org/10.15155/3-00-0000-0000-0000-08C04M

Proceedings of the 9th International Workshop on Computational Linguistics for Uralic Languages, pages 104–108

present in the ENC, and experiment with continuing the training of either only the embeddings or all model parameters.

2 Related Work

Adapting a tokenizer to a new domain or language has been mainly done in two ways: modifying an existing tokenizer or training a new tokenizer on the domain data. The main focus of previous works has been on the embedding initialization methods for new or extended vocabulary, which is not needed in cases of vocabulary pruning.

The vocabulary pruning has been previously explored by Abdaoui et al. (2020). The main motivation was that mBERT (Devlin et al., 2019), for instance, allocates more than 51% of its parameters to the embeddings layer, yet only a fraction of the vocabulary is used for a single given language. Accordingly, the proposed approach is to create monolingual models from multilingual mBERT by shrinking the vocabulary of the original model. To select the tokens to keep for a given monolingual model, the authors collected token frequency statistics from the Wikipedia of the target language, and used these frequencies to filter out the tokens that appeared in less than 0.05% lines. As a result, the monolingual models retained up to 23.8% of the original vocabulary.

Tokenizer extension for BPE models is usually done by first training a new tokenizer and then adding non-overlapping tokens to the existing tokenizer's vocabulary (Csaki et al., 2024; Gee et al., 2022, etc). Csaki et al. (2024) investigated extending an existing tokenizer and found that a correctly implemented vocabulary extension does not negatively affect downstream performance. Tejaswi et al. (2024) also studied the vocabulary extension of LLMs, finding that a larger extension requires more pre-training data for optimal results.

Gee et al. (2022) introduced a method for fast vocabulary transfer (FVT) to train a domain-specific tokenizer. The embeddings for tokens shared between the new and original tokenizers were copied. The embeddings for new tokens were obtained by averaging their sub-token embeddings from the original tokenizer. The model was then further pre-trained on in-domain data using the masked language modeling (MLM) loss before fine-tuning on target tasks. Dagan et al. (2024) applied the FVT (Gee et al., 2022) for LLMs and found that the tokenizer choice impacts the effectiveness and downstream performance of LLMs. Specifically, they found that adapting the model to a new tokenizer requires tens of billions of tokens of retraining to outperform the original tokenizer. While our training ENC training corpus is far smaller, containing only few billions of tokens, we are experimenting with encoder models that are much smaller than LLMs.

3 Methodology

The overall methodology of optimizing a model's vocabulary to Estonian entails two main steps: 1) modifying the content of the vocabulary and adjusting the embeddings accordingly, and 2) continuing the training of the whole model to align it better with the new vocabulary. As the base multilingual model we select mDeBERTa v3—the multilingual version of DeBERTa V3 (He et al., 2023)—the model which is considered the SOTA encoder model at the time of the writing.

We compare two methods to modify the vocabulary of the mDeBERTa v3 multilingual language model for Estonian. The first method involves training a new tokenizer on the Estonian National Corpus (ENC), while in the second method we simply prune the model's original vocabulary to remove tokens that are not used in the tokenized ENC.

Retraining the Tokenizer We retained all the original settings (such as special tokens and pre- and post-processing steps) from the base mDeBERTa v3 model and retrained the underlying SentencePiece tokenizer. For new tokens introduced by the retrained tokenizer, we initialized their embeddings using the mean of the original embedding matrix, while for tokens present in both the original and new tokenizers, we copied the existing embeddings. We also adjust the token to id mapping and resize the embedding matrix.

To align the newly initialized embedding vectors with the rest of the model, we first train the model on the training corpus with only the embeddings unfrozen using the masked language modeling (MLM).

Tokenizer Pruning The pruning process starts with applying the existing tokenizer to the training data. Then we observed what tokens in the vocabulary never appear in our training data and removed them from the vocabulary. After that the token to id mapping was adjusted and the embedding matrix was rearranged and resized. Since no new tokens

were added, we retained the original embeddings for the remaining tokens.

In our experiments, we use an approach similar to Abdaoui et al. (2020) with two key differences. Firstly, we do not use a frequency threshold, but rather keep all the tokens that do appear in the language-specific data. This results in our model retaining approximately 67% of the original vocabulary. Secondly, we employ a larger data source (that also includes Wikipedia)—the Estonian National Corpus. Both differences are aimed at maximizing the vocabulary coverage.

Continuous Training with LoRA To simulate continual training and assess the model's adaptability after vocabulary modification, we applied Low-Rank Adaptation (LoRA) (Hu et al., 2021) training with MLM objective to both models.

4 Experimental Setup

Training Data For training the new tokenizer, and training and validation of the MLM objective, we employed the Estonian National Corpus (ENC). [2] The corpus contains approximately 16M documents with documents coming from different domains such as old and contemporary literature, academic texts, Wikipedia pages and discussions, as well crawled web pages. We performed light deduplication on the corpus resulting in ca 3.4B tokens and randomly split it into train, validation, and test, with both validation and test sets containing 1% of the documents.

Models Developed In our experiments we employ the base version of mDeBERTa V3 as our base model, and apply the previously described approaches—tokenizer retraining and pruning—to it. For tokenizer retraining, we settle for 32K tokens in the final vocabulary, and train the new tokenizer using the train split of the ENC. Meanwhile, for pruning we collect the statistics on the appearance of tokens in the base model vocabulary in the ENC train split, then we remove all tokens that never appear in the data. This results in the vocabulary size of approximately 169K tokens. For both approaches we resize the embedding matrix and rearrange the corresponding vectors, while for tokenizer retraining we initialize the vectors that has not previously appeared in the base model vocabulary using the mean of the embedding matrix.

[2] https://doi.org/10.15155/3-00-0000-0000-0000-08C04M

The models were trained on the University High-Performance Cluster (University of Tartu, 2018) using up to two A100 80GB GPUs.

Embedding training For both approaches, we tuned the embeddings for a single epoch on sequences of 128 tokens in half-precision. The number of devices, per device batch size, and gradient accumulation steps were configured so that the global batch size was 3092. The warm-up ratio was set to 0.05.

Continuous training with LoRA Most of the training parameters remain the same for LoRA continuous training, except for the learning rate which we set to be 1e-3. The LoRA itself was configured with a rank of 4 for the update matrices, using a scaling factor (α) of 32. A dropout rate of 0.1 was applied to the LoRA layers to prevent overfitting. The adaptation was applied to the attention mechanism components (query, key, and value matrices) as well as the dense feed-forward layers. No bias parameters were updated during training.

Fine-tuning on NER Intuitively, a downstream task where the model has to produce classification scores for individual tokes in the input is affected the most by the vocabulary modification. The most common type of such task is likely Named Entity Recognition (NER). For the Estonian language EstNER (Sirts, 2023) is the most comprehensive NER dataset. It contains 46K sentences annotated with 11 entity classes. To assess the performance of the modified models, we fine-tuned models on EstNER for 50 epochs in half-precision with a global batch size of 64. For each model version we repeated the process three times and recorded the highest achieved F1 score in each run. We report the mean and the standard deviation over the three runs.

Model	# Params	Vocab Size	Tok per Word
EstBERT	124M	50K	1.82
XLM-RoBERTa base	278M	250K	2.04
TartuNLP Est-RoBERTa	278M	250K	2.04
EMBEDDIA Est-RoBERTa	116M	40K	1.69
mDeBERTa base	279M	250K	2.23
mDeBERTa base Tuned	110M	32K	1.75
mDeBERTa base Pruned	215M	169K	2.23

Table 1: Statistics on vocabulary size, number of parameters, and tokens per word (estimated on the validation split of the ENC) for related models.

Model	F1 Score (Mean ± Std)	MLM Accuracy
EstBERT	75.72 ± 0.19	-
XLM-RoBERTa base	80.66 ± 0.37	-
TartuNLP EstRoBERTa	81.37 ± 0.28	-
Embeddia EstRoBERTa	$\mathbf{83.77} \pm 0.24$	-
mDeBERTa-base	80.96 ± 0.19	-
mDeBERTa-base → Tuned Embeddings	76.40 ± 0.23	15.86
mDeBERTa-base → Tuned Embeddings → LoRA	77.58 ± 0.47	29.74
mDeBERTa-base → Pruned	80.62 ± 0.12	-
mDeBERTa-base → Pruned → Tuned Embeddings	80.45 ± 0.22	25.84
mDeBERTa-base → Pruned → LoRA	80.62 ± 0.10	38.42

Table 2: EstNER Evaluation F1 and ENC MLM Accuracy scores (excluding baseline models).

Evaluation Metrics We evaluate the tokenization efficiency by calculating the token per word ratio for different tokenizers. We measure the performance of the models on MLM objective using word prediction accuracy. To evaluate the downstream NER task we use the F1 score.

5 Results

In addition to the DeBERTa baseline, we also compare with various other models, including both Estonian-specific EstBERT (Tanvir et al., 2020), XLM-RoBERTa base (Conneau et al., 2020), an Estonian-specific EstRoBERTa finetuned from the XLM-RoBERTa[3] and another EstRoBERTa model trained from scratch.[4]

Tokenizer efficiency We first present the impact on the models' size and tokenizer efficiency in Table 1. We observe that adopting the smaller 32K language-specific vocabulary (mDeBERTa base Tuned) leads to approximately 60% reduction in the number of parameters and 20% reduction in tokens per word. Meanwhile, simply pruning the vocabulary (mDeBERTa base Pruned) results in ca 23% reduction in the number of parameters.

Tokenizer Optimization Results The models with optimized vocabulary MLM accuracy and the downstream NER task F1-scores are shown in Table 2. The top part shows the results for the baseline mDeBERTa base and the comparison models. The baseline mDEBERTa is in line with the multilingual XLM-RoBERTa, but little bit worse than Estonian-specific RoBERTa models. The middle section of the Table 2 shows the results on the models with newly created 32K vocabulary both only after the embedding tuning and then after training continuation with LoRA. While training continuation with LoRA substantially improves the MLM accuracy, replacing the tokenizer led to a substantial decrease in NER performance, with the model average F1 score being below both the baseline multilingual models and two out of three language specific models. The bottom section of the Table 2 shows the results for the models with vocabulary pruning. Again, continuation with the LoRA training improves the MLM accuracy, while the NER results are in the same range with the baseline. The embedding tuning and LoRA training took approximately 120 GPU hours each with the pruned model taking longer due to the large vocabulary size.

6 Discussion

While replacing the vocabulary of the mDeBERTa model with a smaller Estonian-specific vocabulary led to more efficient input tokenization, the results on the downstream NER task suffered even after both embedding layer training and subsequent full model training with LoRA. First, suboptimal embedding initialization approach likely played a role in the observed outcome. Secondly, it is likely that a single epoch of embedding tuning was insufficient to match the performance of the base model. The subsequent LoRA MLM training resulted in slightly reducing the gap between the base model and the model with the retrained tokenizer, however it also remained insufficient to recover the original model's performance. We presume that training for longer, both the embeddings and the LoRA parameters, would further reduce that gap.

In contrast, we observe that the vocabulary pruning has no observable negative effect on the downstream task. Meanwhile, tuning of the embeddings appears to have little to no effect on the downstream task, which suggests that such tuning is redundant

[3] https://huggingface.co/tartuNLP/EstRoBERTa
[4] https://huggingface.co/EMBEDDIA/est-roberta

in case of pruning. Surprisingly and similarly to embeddings tuning, continued training with LoRA had no observable benefit for the pruned model, despite the gains in the MLM accuracy.

Finally, we observed that the relation between the MLM accuracy and F1 on NER is not transparent. While we acknowledge that MLM accuracy scores with different vocabularies are not directly comparable, the absence of the effect on the NER result in the presence of a notable improvement in the MLM accuracy in the pruned model is puzzling.

7 Conclusion

In this study, we explored two options for optimizing the vocabulary of a multi-lingual model for the Estonian language. In summary, we found that replacing the tokenizer with a retrained language-specific version noticeably degrades model performance on the downstream NER task, and one epoch of embedding layer training on a 3.4B word corpus did not suffice to restore it. While LoRA offers efficient way for further training continuation, a single epoch was insufficient to mitigate the negative impact of the tokenizer replacement. On the other hand, pruning unused tokens proved to be an effective method to reduce vocabulary size without compromising performance.

Acknowledgments

This research was supported by the Estonian Research Council Grant PSG721 and Estonian Language Technology Program Grant EKTB104.

References

Amine Abdaoui, Camille Pradel, and Grégoire Sigel. 2020. Load What You Need: Smaller Versions of Mutililingual BERT. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 119–123, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Zoltan Csaki, Bo Li, Jonathan Li, Qiantong Xu, Pian Pawakapan, Leon Zhang, Yun Du, Hengyu Zhao, Changran Hu, and Urmish Thakker. 2024. Sambalingo: Teaching large language models new languages. *arXiv preprint arXiv:2404.05829*.

Gautier Dagan, Gabriel Synnaeve, and Baptiste Roziere. 2024. Getting the most out of your tokenizer for pre-training and domain adaptation. In *Forty-first International Conference on Machine Learning*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torroni. 2022. Fast Vocabulary Transfer for Language Model Compression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 409–416, Abu Dhabi, UAE. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *Preprint*, arXiv:2111.09543.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for Russian language. *arXiv preprint arXiv:1905.07213*.

Robert Mroczkowski, Piotr Rybak, Alina Wróblewska, and Ireneusz Gawlik. 2021. HerBERT: Efficiently Pretrained Transformer-based Language model for Polish. In *Proceedings of the 8th Workshop on Balto-Slavic Natural Language Processing*, pages 1–10.

Kairit Sirts. 2023. Estonian Named Entity Recognition: New Datasets and Models. In *Proceedings of the 24th Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 752–761, Tórshavn, Faroe Islands. University of Tartu Library.

Hasan Tanvir, Claudia Kittask, and Kairit Sirts. 2020. EstBERT: A Pretrained Language-Specific BERT for Estonian. *Preprint*, arXiv:2011.04784.

Atula Tejaswi, Nilesh Gupta, and Eunsol Choi. 2024. Exploring Design Choices for Building Language-Specific LLMs. *arXiv preprint arXiv:2406.14670*.

University of Tartu. 2018. UT Rocket.