A Comparison of Different Tokenization Methods for the Georgian Language

Beso Mikaberidze[†], **Teimuraz Saghinadze**[†], **Guram Mikaberidze**[‡], **Raphael Kalandadze**[†], **Konstantine Pkhakadze**[¶], **Josef van Genabith**[*], **Simon Ostermann**[*], **Lonneke van der Plas**[**], **Philipp Müller**[*]

[†]MICM Georgia, [‡]UWYO USA, [¶]GTU Georgia, [**]IDIAP Switzerland, [*]DFKI Germany

beso.mikaberidze@gmail.com, philipp.mueller@dfki.de

Abstract

While the impact of tokenization on language modeling is well-researched in richly resourced languages, fewer studies on this topic exist for challenging low-resource languages. In this work, we present the first systematic evaluation of tokenization methods for Georgian, a low-resource language with high morphological complexity. We compare standard subword tokenizers, such as WordPiece, Byte Pair Encoding, SentencePiece with Unigram, and a recently proposed token-free approach. We also investigate the multilingual BERT tokenizer (mBERT), which includes Georgian. In addition to these different classes of tokenization algorithms we also evaluate the impact of different vocabulary sizes, a key parameter for subword tokenizers. We evaluate the performance of all tokenizers on masked language modeling and on four downstream tasks: part-of-speech tagging, named entity recognition, toxicity detection, and sentiment analysis. We observe that larger vocabulary sizes for subword tokenizers generally lead to better performance across most tasks, with a notable exception in the toxicity detection task, where finer subword granularity is more effective. For the remaining tasks, pre-training tokenizers on Georgian text consistently yield better results compared to mBERT. Additionally, the token-free method is consistently outperformed by all other tokenizers. Taken together, our comprehensive evaluation of tokenizers will be highly valuable in making informed tokenization choices in future language model developments for Georgian.

1 Introduction

Tokenization is a fundamental process in most natural language processing (NLP) tasks that involves breaking down a text into smaller units called *tokens*. It is one of the first processes conducted in most approaches and is particularly crucial for low-resource languages. Tokenization gains further importance in morphologically complex languages where multiple types of prefixes and suffixes simultaneously modify the meaning of a word, making it vital to split each word into meaningful pieces. That is why different tokenization methods have been investigated in languages such as Turkish (Toraman et al., 2023), Arabic (Alyafeai et al., 2023), or Korean (Park et al., 2020). Studies on these languages have shown that appropriate tokenization can significantly enhance model performance, with subword-level tokenization often providing a good balance between capturing linguistic nuances and managing sequence lengths.

In contrast, no comprehensive study of tokenization has been conducted for any of the languages from the Kartvelian family to which Georgian belongs. The Kartvelian family has no known relation to any other language group. It consists of four languages, all spoken in Georgia, with its first split dating back to the 20-22th century BC (Gavashelishvili et al., 2023). Georgian, the official language of Georgia, serves as a common language for all Kartvelian speakers. The language is phonetic and is written in its unique alphabet, one of the world's approximately 15 base alphabetical systems. Georgian, a low-resource language with complex morphology, has seen limited progress in NLP research, which remains in its early stages. Existing studies have primarily focused on data curation (Beridze et al., 2017; Stefanovitch et al., 2022a) and syntactic and morphological analysis (Kapanadze, 2019; Kardava et al., 2017; Lobzhanidze, 2022) rather than tokenization. Conducting a comprehensive evaluation of tokenizers for Georgian provides a solid foundation for future research on building effective Georgian language models, addressing its unique linguistic challenges, and improving NLP applications.

In our work, we address this need by, for the first time, systematically evaluating different tokenizers in Georgian for language modeling and on a set of four downstream tasks. In particular, we evaluate four tokenization techniques: WordPiece (Song et al., 2021), Byte Pair Encoding (Sennrich et al., 2016), SentencePiece with Unigram (Kudo and Richardson, 2018, Kudo, 2018), and a token-free method (Xue et al., 2022). With these tokenizers, we train a scaled-down BERT (Devlin et al., 2018) architecture on a substantial Georgian language corpus and fine-tune it on four downstream applications: sentiment analysis, toxicity detection, named entity recognition, and part-of-speech tagging. In addition, we investigate various vocabulary sizes by training different-sized tokenizer models, identifying optimal strategies tailored to Georgian's morphological characteristics. Our results indicate that (1) subword tokenization approaches trained on Georgian pretraining corpora are superior to the token-free approach as well as multilingual BERT's WordPiece tokenizer, and (2) that larger vocabulary sizes tend to improve performance. The main exception is the toxicity detection task, where tokenizers with finer granularity perform better. These include multilingual BERT's WordPiece with its smaller vocabulary as well as the smaller vocabulary versions of the subword tokenizers. With our approaches, we set a new state of the art on the recently introduced toxicity detection dataset by Lashkarashvili and Tsintsadze (2022).

The source code developed in this study is available online[1].

2 Related Work

2.1 Tokenizers in Language Modelling

We distinguish three major categories of tokenizers: word-level, subword-level, and token-free (character/byte-level tokenizers).

Word-level tokenizers take all distinct words in the corpus as tokens, which results in large vocabularies that are, however, still rarely exhaustive. While not requiring specific training, such tokenizers often suffer from numerous out-of-vocabulary cases (Luong et al., 2015).

Subword-level tokenization is the most common tokenization technique for modern language models. Such tokenizers are trained and selectively combine characters, subwords, and words. Words that are rarely used are usually split into smaller units, resulting in smaller vocabulary sizes at better coverage and fewer out-of-vocabulary cases.

GPT 2, 3 and RoBERTa (Radford et al., 2019, Brown et al., 2020, Liu et al., 2019) utilize a Byte Pair Encoding (BPE) tokenization method (Sennrich et al., 2016). BERT and ELECTRA (Devlin et al., 2018, Clark et al., 2020) use a variant of the BPE, the WordPiece tokenization method (Song et al., 2021). XLM-RoBERTa, XLNet, and T5 (Conneau et al., 2020, Yang et al., 2019, Raffel et al., 2020), all rely on SentencePiece (Kudo and Richardson, 2018) with the Unigram algorithm Kudo (2018).

Token-free approaches treat all distinct characters or bytes in the corpus as tokens, resulting in a small vocabulary and no out-of-vocabulary cases, but also significantly longer input sequences and less meaningful individual tokens. Byte-level tokenizers have been shown to be competitive with their subword-level counterparts but usually need more training time (Xue et al., 2022).

2.2 Tokenization for Morphologically Rich and Low-Resource Languages

Toraman et al. (2023) show that for languages with rich morphology, the choice of tokenizer can significantly affect model performance. Word-level tokenization often struggles due to the large number of possible word forms, whereas subword-level tokenizers and token-free approaches can provide more flexibility and robustness by capturing meaningful subunits and handling out-of-vocabulary words effectively. Similarly, Park et al. (2020) discuss the importance of appropriate tokenization for Korean, a language with agglutinative morphology. They highlight how different tokenization strategies, such as character-level and subword-level, affect the performance of NLP models on diverse tasks and show that subword-level tokenization strikes a balance between capturing linguistic nuances and maintaining manageable sequence lengths. Alyafeai et al. (2023) examine how different tokenization methods perform on Arabic text classification tasks. Given the rich morphology and script variations, they show that tokenizers that can effectively handle these complexities are required. Subword-level tokenization, in particular, has been shown to provide better performance by capturing root and pattern morphemes.

2.3 Georgian Natural Language Processing

Georgian, a highly inflectional and agglutinative language with complex morphology, poses unique challenges for tokenization. Kartvelian, primarily

[1] https://git.opendfki.de/philipp.mueller/icnlsp24

spoken in Georgia, has no known relation to any other language groups, making it one of the world's primary language families.

Research on Georgian NLP is still in its early stages and to the best of our knowledge, no existing study focuses on tokenization methods. The majority of research has concentrated on data curation (Beridze and Nadaraia, 2009; Doborjginidze and Lobzhanidze, 2016; Fkhakadze et al., 2017; Beridze et al., 2017; Stefanovitch et al., 2022a) and automated syntactic and morphological analyzers (Kapanadze et al., 2019; Kapanadze, 2019; Kardava et al., 2017; Lobzhanidze, 2022). Some studies have trained models for downstream applications (Khachidze et al., 2016; Lashkarashvili and Tsintsadze, 2022; Stefanovitch et al., 2022a), using standard tokenization techniques without exploring the impact of tokenizers on the model's performance. Several papers (Pires et al., 2019, Conneau et al., 2020) with pre-trained multilingual language models provide subword-level tokenizers containing Georgian tokens. However, the current state of research indicates a gap in understanding how different tokenizers would perform for Georgian.

While subword-level tokenization has proven effective for large language models, even for morphologically rich languages like Turkish, Korean, and Arabic, the question of which subword-level algorithm would be most effective for Georgian remains open. The competitiveness of the token-free approach is also uncertain.

3 Data

In the following Section we present pre-training datasets and downstream task datasets used in our study.

3.1 Pre-training Datasets

In this work, we ensured a comprehensive coverage of the various styles and contexts of Georgian. We used three primary corpora to pre-train our tokenization models: Wikipedia [2], Leipzig (2016) [3], and CorpusGE (Fkhakadze et al., 2017). Wikipedia and Leipzig provide extensive text data across various domains, ensuring diverse language coverage. CorpusGE, a high-quality text corpus, was collected over four years from well-known Georgian media pages. Following previous work on Maltese (Micallef et al., 2022), we employed one million words from these corpora to pre-train our models.

	NER	POS	TOXD	SA
Epochs	10	30	10	10
Max. length	512	512	512	512
Batch size	384	24	192	192
Learning rate	2e-5	2e-5	2e-5	2e-5
Train Size	90,000	2,000	8,000	2,500
Val. Size	90,000	250	1,000	850
Test Size	92,000	250	1,000	850

Table 1: Training details for the four different tasks: Named-Entity Recognition (NER), Part-of-Speech Tagging (POS), Toxicity Detection (TOXD), and Sentiment Analysis (SA). In train and test sizes, we provide labeled word counts for token classification and labeled sentence counts for text classification tasks. **Exception:** Maximum length for token-free ByT5 is equal to 2048.

3.2 Downstream Tasks

To assess the performance of different tokenization methods, we focused on four language understanding tasks: two for text classification and two for token classification. We present an overview over downstream task dataset sizes in Table 1.

Named Entity Recognition (NER) Named Entity Recognition is a token classification task that identifies person, organization, or location names in the text. We utilized the Wikiann (pan-x) multilingual benchmark (Pan et al., 2017), a comprehensive dataset that includes Georgian. This benchmark, which consists of approximately 30,000 Georgian sentences and roughly 90,000 labeled words per train, validation, and test splits, provides a thorough dataset for NER.

Part-of-Speech Tagging (POS) Part-of-Speech Tagging is a token classification task that detects parts of speech with respect to each word in a sentence, such as nouns, verbs, adjectives, etc. We employed the Universal Dependencies dataset (Nivre et al., 2020) for Georgian, which contains approximately 2,500 words and 152 sentences. The dataset was split, with 10% used for validation and 10% for testing. We also provide the percentage break-

[2] https://dumps.wikimedia.org/backup-index.html

[3] https://wortschatz.uni-leipzig.de/en/download/Georgian

down of fourteen imbalanced class distributions: Noun (29%), Punc (14%), Adj (13%), Verb (9%), Pron (8%), Post (7%), Conj (6%), Adv (6%), Aux (3%), Part (2%), Prop (2%), Num (1%), VerbalAdj (0.3%), VerbalNoun (0.1%).

Toxicity Detection (TOXD) Toxicity detection is a text classification task identifying harmful or toxic comments in online discussions. For this task, we used a dataset provided by Lashkarashvili and Tsintsadze (2022). This data was gathered from Georgian online discussion forums and manually annotated for toxicity. The dataset comprises 10,000 sentences, divided into 46% toxic and 54% non-toxic samples. We split the train, validation, and test datasets as follows: 80%, 10%, and 10%.

Sentiment Analysis (SA) Sentiment Analysis is a text classification task that determines the emotional tone of the text. For this task, we used the first publicly released annotated sentiment dataset for Georgian (Stefanovitch et al., 2022b), referred to as Georgian Sentiment Snippets (GSS). This dataset contains approximately 4K text snippets, each manually annotated by multiple annotators using a four-tier scale: positive (33.5%), neutral (41.0%), negative (18.1%), and mixed (7.2%). The dataset is split into training, validation, and test sets with the following proportions: 60%, 20%, and 20%. This annotated dataset provides a robust resource for training and evaluating sentiment analysis models.

4 Approach

We first discuss the different tokenizers we compare in our study and subsequently present the training procedure we utilized.

4.1 Tokenizers

In this study, we compare various tokenization methods for Georgian. We focus on subword-level tokenizers, including WordPiece, Byte Pair Encoding (BPE), and SentencePiece with Unigram. Additionally, we explore the byte-level token-free approach ByT5, assessing its performance relative to traditional subword-level tokenizers.

Byte Pair Encoding (BPE) This approach was initially introduced for data compression (Sennrich et al., 2016). BPE minimizes the total number of symbols (characters or bytes) needed to represent the original text. First, the data is split into individual symbols. Then, the most frequent adjacent pairs of symbols are consecutively merged until the desired vocabulary size is reached. In this study, we employ BPE, which considers every distinct byte as its initial vocabulary.

WordPiece (WP) WordPiece (Song et al., 2021) is a variant of the BPE method. The primary difference lies in the merge rule, which is based on likelihood rather than solely on frequency. Specifically, the algorithm prioritizes token pairs that have a higher joint probability of how frequently the tokens appear together compared to how frequently they appear separately. This method aims to retain more meaningful linguistic units, potentially providing a more nuanced tokenization. However, training requires more computational resources due to the complexity of calculating these probabilities.

SentencePiece with Unigram (SP-U) The SentencePiece (Kudo and Richardson, 2018) is a tool that implements both the BPE and Unigram (Kudo, 2018) algorithms. This approach enables the tokenization of raw text strings without the need for preprocessing, such as whitespace splitting, making it particularly effective for languages without clear word boundaries.

The Unigram algorithm, employed within the SentencePiece framework, operates in two stages. First, it populates its vocabulary with a large number of tokens similar to BPE, but for searching the most frequent substrings, it uses the enhanced suffix array algorithm. Second, it decreases the vocabulary to the desired size. The Unigram model iteratively prunes the least likely tokens based on their probability contribution to the corpus, leveraging the expectation maximization (EM) algorithm.

Token-Free Byt5 This approach treats all distinct bytes in the corpus as tokens. Xue et al. (2022) used this approach and have increased the number of transformer parameters at the expense of a large number of discarded vocabulary parameters. They have been shown to be competitive with their subword-level counterparts.

Multilingual BERT To provide a comprehensive evaluation, we compare our pre-trained tokenizers with out-of-the-box multilingual BERT's (Pires et al., 2019) WordPiece tokenizer, containing 700 Georgian tokens. This comparison allows us to assess the effectiveness of our tokenizers against the established multilingual model.

4.2 Training Procedure

Tokenizer Training As the vocabulary size is a critical factor for the subword tokenizers, we ensure its optimization for each method. Each subword-level tokenizer was trained to generate vocabularies of four different sizes (8k, 16k, 32k, and 64k), ensuring optimal performance for the BERT model. All the tokenizers were adjusted to accommodate BERT's special tokens and post-processing requirements.

BERT Integration For the integration with BERT, we followed related studies (Toraman et al., 2023; Xue et al., 2022) and utilized a scaled-down architecture. These studies indicated that differences between tokenizers are more pronounced with smaller language models. Smaller models also have the advantage of faster training, allowing us to run more evaluations than would be possible with larger-scale models. For our scaled-down BERT model, we used the following configuration, consistent across all our experiments: Hidden size: 512; Number of hidden layers: 8; Number of attention heads: 8; Intermediate-size: 3072; Max position embeddings: 512 for subword-level tokenizers, 2048 for token-free approach.

Pre-training Setup The pre-training corpus, as detailed in Sec. 3, comprises 1 million tokens from high-quality Georgian text sources. Pre-training was conducted by training multiple BERT models sufficiently long to achieve stable training and evaluation loss plots. BERT models were pre-trained using only the Masked Language Modeling (MLM) task, with the following aspects deviating from the original BERT configuration. We made use of dynamic masking adopted from RoBERTa, set the training epochs to 30, the batch size to 264, and employed mixed precision training.

Finetuning and Evaluation The pre-trained BERT models were finetuned on four downstream language understanding tasks: Named Entity Recognition (NER) and Part-of-Speech (POS) tagging for token classification, and Sentiment Analysis and Toxicity Detection for text classification. Details on these tasks and their corresponding datasets are provided in Section 3. Each language model was finetuned 26 times, and evaluation results were averaged across these runs to ensure stability and robustness. Performance was evaluated using four metrics: accuracy, f1 score, precision, and recall. These metrics provide a comprehensive view of the models' effectiveness across the various tokenization methods.

		ByT5	mWP	BPE	WP	SP-U
MLM	**acc**	0.423	0.564	0.613	0.616	**0.617**
NER	**acc**	0.800	0.902	0.925	0.927	**0.930**
	f1	0.552	0.758	**0.797**	0.794	0.787
	pre	0.565	0.744	0.781	**0.783**	0.774
	rec	0.539	0.774	**0.813**	0.806	0.800
TOXD	**acc**	0.879	**0.955**	0.917	0.933	0.941
	f1	0.866	**0.952**	0.911	0.928	0.937
	pre	0.890	**0.948**	0.912	0.923	0.929
	rec	0.843	**0.957**	0.910	0.933	0.945
POS	**acc**	0.699	0.889	0.900	0.905	**0.915**
	f1	0.045	0.709	0.817	**0.824**	0.820
	pre	0.028	0.670	0.788	**0.795**	0.790
	rec	0.121	0.754	0.849	**0.856**	0.852
SA	**acc**	0.493	0.588	0.672	0.668	**0.675**
	f1	0.472	0.558	0.642	0.641	**0.647**
	pre	0.470	0.535	0.637	0.649	**0.663**
	rec	0.493	0.588	0.672	0.668	**0.675**

Table 2: Performance of different tokenizers across various NLP tasks in terms of accuracy, f1 score, precision, and recall. Tokenizers: ByT5, multilingual BERT's WordPiece (mBERT), Byte Pair Encoding (BPE), WordPiece (WP), SentencePiece with Unigram (SP-U). Tasks: Masked Language Modeling (MLM), Named-Entity Recognition (NER), Part-of-Speech Tagging (POS), Toxicity Detection (TOXD), and Sentiment Analysis (SA).

5 Results

5.1 Comparing Tokenizers

We present the results of different tokenizers on language modeling and our four downstream tasks in Table 2. All subword tokenizers in this table were trained with a vocabulary size of 64k. For masked language modelling, SentencePiece with Unigram (SP-U) achieves the highest accuracy of 0.617, closely followed by WordPiece (0.616 acc), and BPE (0.613). Both multilingual BERT's WordPiece tokenizer and the token-free ByT5 are

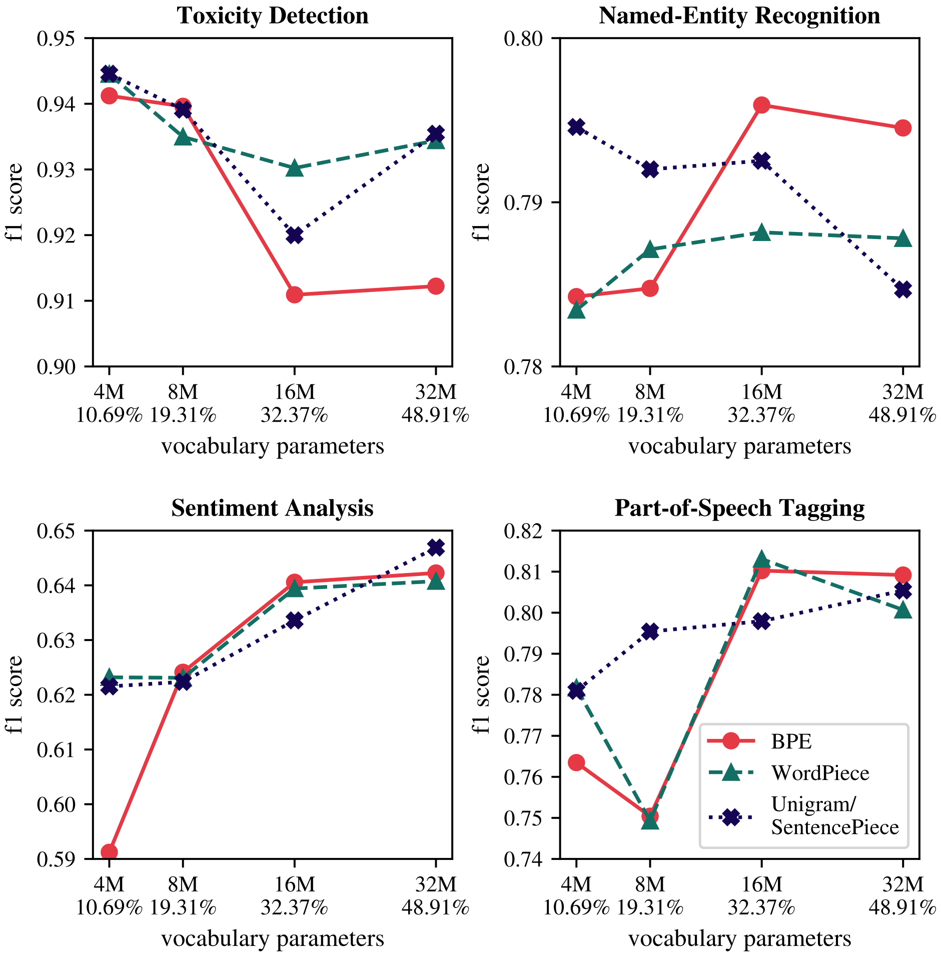

Figure 1: Impact of vocabulary size on the performance of four downstream tasks: Toxicity Detection, Sentiment Analysis, Named-Entity Recognition, and Part-of-Speech Tagging. The x-axis shows the vocabulary size in absolute numbers as well as in proportion to the overall network parameters.

worse by a large margin (0.56 and 0.423 acc, respectively). This general pattern is also present in three out of four downstream tasks. For named entity recognition, part-of-speech-tagging, and sentiment analysis, the subword tokenizers consistently achieve better performance than ByT5 and multilingual BERT's WordPiece tokenizer. The differences between subword tokenizers on downstream tasks are small. When measured in terms of f1, BPE achieves the best performance in named entity recognition (0.797 f1). For part-of-speech tagging, WordPiece achieves the best f1 score of 0.824, and for sentiment analysis, SentencePiece with Unigram is leading with 0.647 f1.

We observed surprisingly bad POS results for the ByT5 tokenizer in terms of f1 (0.045), precision (0.028), and recall (0.121). We conjecture this is because f1, precision, and recall are directly related to the number of correctly predicted positive instances. Because the tokenizer breaks the text into tokens that are too granular or not meaningful for the POS tagging task, combined with a small number of training examples, there is a high number of false positives and false negatives, thereby lowering the aforementioned metrics. Also, our POS tagging benchmark is highly imbalanced and involves a few frequent tags, like nouns and verbs, and many infrequent ones, like rare numerals. Thus, a high accuracy is misleading to some extent as the model performs well on frequent tags while failing on the rare ones.

5.2 Comparing Vocabulary Sizes

We present the results of our vocabulary size experiments in Figure 1. There is a tendency that larger vocabulary sizes lead to better performance. This is clearly the case for both sentiment analysis and part-of-speech tagging. For named entity recognition, the effect of vocabulary sizes is negligible - f1

	Accuracy	**AUC**
SOTA CNN	0.888 ± 0.007	0.942 ± 0.005
Ours (WP 8k)	0.944 ± 0.007	0.944 ± 0.007
Ours (mWP)	$\mathbf{0.959} \pm \mathbf{0.009}$	$\mathbf{0.959} \pm \mathbf{0.009}$

Table 3: Our Toxicity Detection approaches compared with the SOTA by Lashkarashvili and Tsintsadze (2022). We report accuracy and area under curve (AUC), along with standard deviations across CV folds.

scores only vary between 0.78 and 0.80. However, for toxicity detection, the positive connection between vocabulary size and performance is clearly reversed. Here, tokenizers with a higher average split of the words were more effective. This indicates that a finer granularity in tokenization can be beneficial for tasks requiring a nuanced understanding of potentially offensive language. This statement is in line with the previously observed fact that the multilingual BERT (mBERT) tokenizer performs best for toxicity detection. The mBERT tokenizer contains only 700 Georgian tokens, the smallest vocabulary size among the subword-level tokenizers we investigated.

5.3 Comparison with SoTA Approaches

Our scaled-down BERT models (42M parameters) demonstrate strong performance on the Toxicity Detection dataset introduced by Lashkarashvili and Tsintsadze (2022). We employed two tokenization methods for pretraining and fine-tuning: an 8K vocabulary WordPiece and the multilingual BERT WordPiece. For comparability, we followed Lashkarashvili and Tsintsadze (2022) by using stratified 5-fold cross-validation, along with accuracy (ACC) and area under the curve (AUC) as evaluation metrics. Results, presented in Table 3, show that while Lashkarashvili and Tsintsadze (2022) reported an ACC of 0.888 and an AUC of 0.942 for their best-performing CNN model, our approach achieved an ACC of 0.9435 and an AUC of 0.9442 using the 8K WordPiece, and an accuracy of 0.9586 and an AUC of 0.9591 with the multilingual WordPiece, establishing a new state of the art.

6 Discussion

For most tasks, we observed that pre-training tokenizers on a small amount of Georgian text yield better performance than relying on the mBERT tokenizer. This suggests that language-specific pretraining is crucial for achieving optimal results in Georgian NLP tasks. The superior performance of these tokenizers compared to the multilingual WordPiece tokenizer from mBERT (except in toxicity detection) raises questions about the limitations of the latter. Our findings indicate that this multilingual tokenizer may not adequately capture the nuances of highly divergent languages such as Georgian.

Furthermore, our results indicate that ByT5 is not competitive with the other tested methods. We suspect two possible reasons for this. First, each Georgian letter contains 3 bytes, so the LM training input sequences for Georgian are three times longer than for English. Second, in the original ByT5 paper, the authors Xue et al. (2022) increased the number of transformer parameters at the expense of a large number of discarded vocabulary parameters. They increased input sequence length, embedding size, and intermediate layer size. We only increased the input sequence length due to the limited vocabulary parameters, which might be another reason for the suboptimal performance observed.

We found a general trend of improved performance with larger vocabulary sizes for subword tokenizers. This suggests that capturing a wide range of morphological variations is crucial for effective language modeling in Georgian. However, our findings on toxicity detection versus the other downstream tasks also underscore the importance of tailoring tokenization strategies to the specific requirements of each task and dataset.

In our case, the toxicity detection benchmark involves words that are not present in the tokenizer's vocabulary, specifically those that serve as key indicators of toxic content. When a tokenizer encounters these unknown words, it splits them into smaller subword units. This behavior is observed even in tokenizers with large vocabularies. However, LMs using tokenizers with smaller vocabulary sizes are inherently more robust at handling and representing short tokens because their pretraining data mostly contains short tokens. In contrast, LMs using tokenizers with larger vocabularies tend to rely on longer tokens, which can lead to a loss of information when the input is split into less meaningful or less frequent short tokens. We conjecture that this is the reason for why tokenizers with smaller vocabulary sizes perform better in the case of toxicity detection.

The results highlight that a one-size-fits-all approach to tokenization is inadequate, and careful consideration must be given to the nature of the task and especially to the linguistic features of a language.

7 Conclusion

In this study, we explored the impact of various tokenization methods on Georgian language modeling, including subword-level tokenizers, such as BPE, WordPiece, and SentencePiece with Unigram, a pre-trained multilingual BERT tokenizer, and a recently proposed token-free approach ByT5. Each method is evaluated by the performance of a scaled-down BERT architecture on four independent downstream tasks. Our findings suggest that larger vocabulary sizes generally enhance performance across most NLP tasks. However, on the toxicity detection task, tokenizers with finer granularity, like the multilingual mBERT with its smaller vocabulary, performed better. In all the other tasks, language-specific pre-training of tokenizers outperformed mBERT. Interestingly, the token-free approach did not perform competitively, highlighting potential limitations of its applicability to Georgian, our model's architecture, or both.

In the future, we aim to explore the impact of different tokenization strategies on more advanced model architectures, as well as extend this analysis to other Kartvelian languages, which could further our understanding of effective NLP strategies for Georgian and similar languages.

Georgian presents a challenging landscape for NLP due to its complex morphology, limited training data, and sparse research focus. By conducting the first rigorous comparative study of tokenization methods for Georgian, this work lays a foundational reference for future research and development. Given that tokenization is the first step in NLP model training, our study provides valuable insights that can guide researchers and practitioners in building models tailored to the needs of Georgian and specific NLP tasks.

Limitations

While this study provides valuable insights into tokenization methods for Georgian, several limitations should be acknowledged.

Architectural Diversity Our research is limited to a scaled-down BERT. Exploring and experimenting with other LM architectures could potentially yield different results and even trends, and thus, it is essential to consider alternative architectures in future studies.

Language Scope The experiments and analyses conducted in this study are restricted to Georgian. Testing the generalizability of our findings to other languages would provide a broader validation of the tokenization methods. This is particularly important for ensuring the robustness and applicability of our approaches in multilingual contexts.

Downstream Tasks Our study evaluates the tokenization methods on a limited number of downstream tasks. Expanding the range of downstream tasks in future research will help to understand the effectiveness and limitations of the tokenization methods in diverse applications, potentially uncovering further task-specific strengths and weaknesses.

Comparison with Multilingual Models While we compared our tokenizers to the multilingual BERT model, we did not include XLM-RoBERTa and mT5 (Conneau et al., 2020; Xue et al., 2021) tokenizers in our evaluation. Future work should incorporate this and other recent multilingual models to provide a more complete comparison.

Acknowledgements

This work was partially supported by the European Union under Horizon Europe project *"GAIN"* (GA #101078950) and *DisAI - Improving scientific excellence and creativity in combating disinformation with artificial intelligence and language technologies* (GA #101079164), as well as the BMBF-funded project TRAILS (01IW24005).

References

Zaid Alyafeai, Maged S. Al-shaibani, Mustafa Ghaleb, and Irfan Ahmad. 2023. Evaluating various tokenizers for arabic text classification. *Neural Processing Letters*, 55(3):2911–2933.

Marina Beridze, David Nadaraia, and Lia Bakuradze. 2017. Georgian dialect corpus: Linguistic and encyclopedic information in online dictionaries. *Journal of Linguistics/Jazykovedný casopis*, 68(2):109–121.

Marine Beridze and David Nadaraia. 2009. The corpus of georgian dialects. *NLP, Corpus Linguistics, Corpus Based Grammar Research*, page 25.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nino Doborjginidze and Irina Lobzhanidze. 2016. Corpus of the georgian language. In *Proceedings of the XVII EURALEX International Congress*, pages 328–335.

Konstantine Fkhakadze, Merab Chikvinidze, Giorgi Chichua, Davit Kurtskhaliya, and Inga Beridze. 2017. Georgian internet and web corpus.

Alexander Gavashelishvili, Merab Chukhua, Kakhi Sakhltkhutsishvili, Dilek Koptekin, and Mehmet Somel. 2023. The time and place of origin of south caucasian languages: insights into past human societies, ecosystems and human population genetics. *Scientific Reports*, 13(1):21133.

Oleg Kapanadze. 2019. Parsing the less-configurational georgian language with a context-free grammar. *Proceedings of the Language Technologies for All (LT4All), European Language Resources Association (ELRA), Paris, UNESCO Headquarters*, pages 342–345.

Oleg Kapanadze, Gideon Kotzé, and Thomas Hanneforth. 2019. Building resources for georgian treebanking-based nlp. In *International Tbilisi Symposium on Logic, Language, and Computation*, pages 60–78. Springer.

Irakli Kardava, Nana Gulua, Jemal Antidze, and Beka Toklikishvili. 2017. Morphological synthesis and analysis of georgian words.

Manana Khachidze, Magda Tsintsadze, and Maia Archuadze. 2016. Natural language processing based instrument for classification of free text medical records. *BioMed research international*, 2016(1):8313454.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Nineli Lashkarashvili and Magda Tsintsadze. 2022. Toxicity detection in online georgian discussions. *International Journal of Information Management Data Insights*, 2(1):100062.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Irina Lobzhanidze. 2022. Computational modeling. In *Finite-State Computational Morphology: An Analyzer and Generator for Georgian*, pages 117–166. Springer.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Kurt Micallef, Albert Gatt, Marc Tanti, Lonneke van der Plas, and Claudia Borg. 2022. Pre-training data quality and quantity for a low-resource language: New corpus and BERT models for Maltese. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 90–101.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Kyubyong Park, Joohong Lee, Seongbo Jang, and Dawoon Jung. 2020. An empirical study of tokenization strategies for various Korean NLP tasks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 133–142, Suzhou, China. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Xinying Song, Alex Salcianu, Yang Song, Dave Dopson, and Denny Zhou. 2021. Fast WordPiece tokenization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2089–2103, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicolas Stefanovitch, Jakub Piskorski, and Sopho Kharazi. 2022a. Resources and experiments on sentiment classification for georgian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1613–1621.

Nicolas Stefanovitch, Jakub Piskorski, and Sopho Kharazi. 2022b. Resources and experiments on sentiment classification for Georgian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1613–1621, Marseille, France. European Language Resources Association.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinüç, and Oguzhan Ozcelik. 2023. Impact of tokenization on language models: An analysis for turkish. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(4).

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. *XLNet: generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA.