

Post-Processing Tokenizer Adaptation Under a Fixed Vocabulary Budget: Compression–Performance Trade-offs Across Models and Tasks

Zhaoxuan Jiang

School of Computer Science, Faculty of Engineering and IT, University of Technology Sydney, Sydney, NSW 2007, Australia

ABSTRACT

Subword tokenization is a key component of modern language models, but general-purpose tokenizers are often inefficient for domain-specific text. Recent work has proposed post-processing replacement methods that adapt a tokenizer by replacing low-frequency subwords with high-frequency words under a fixed vocabulary constraint. However, it remains unclear whether this strategy generalizes across different model architectures and tasks. In this paper, we study a fixed-vocabulary post-processing replacement strategy on two model architectures, GPT-2 and RoBERTa, and three text classification datasets: IMDB, SST-2, and AG News. We evaluate multiple replacement budgets and examine the trade-off between token reduction and classification accuracy. Our results show that the method is highly setting-dependent. GPT-2 maintains stable accuracy under moderate compression, while RoBERTa is more sensitive to replacement and generally shows lower accuracy than the baseline. Across datasets, SST-2 is the most sensitive to token changes, whereas AG News is relatively robust but gains little compression. These findings suggest that lightweight tokenizer adaptation is possible, but its effectiveness depends strongly on model architecture and task characteristics, and compression alone is not sufficient to predict downstream behavior.

KEYWORDS

Tokenization adaptation; BPE; Compression-performance trade-off; Domain adaptation; GPT-2; RoBERTa.

1 Introduction

Subword tokenization is a basic component of modern language models. Methods such as byte pair encoding (BPE) are widely used because they balance vocabulary size and coverage by splitting rare or complex words into smaller units. However, a tokenizer trained on general-purpose text may be suboptimal when applied to a specific domain, where frequent task-relevant words are still broken into multiple tokens. This can increase sequence length and may also weaken useful semantic cues. Recent work has shown that token compression is not only an efficiency issue, but is also related to downstream model performance. This motivates the study of tokenizer adaptation methods that improve tokenization quality without retraining a model from scratch. Among these methods, AdaptBPE proposes a lightweight post-processing strategy that replaces low-frequency subwords with more useful units while keeping the vocabulary size fixed. Its main advantage is compatibility with pretrained models, since the tokenizer can be adapted without changing model parameters. However, the behavior of this strategy is still not well understood in standard text classification settings. Existing evidence mainly focuses on limited architectures and tasks, leaving several practical questions open: whether the method transfers across model types, whether its impact differs across datasets and task styles, and how much replacement can be introduced before performance starts to drop.

To answer these questions, we evaluate a fixed-vocabulary post-processing replacement strategy on GPT-2 and RoBERTa using IMDB, SST-2, and AG News. We focus on the trade-off between token reduction and classification accuracy, and examine how this trade-off changes across architectures, datasets, and replacement budgets.

2 Related Work

Subword tokenization has become a standard component of modern NLP systems because it balances vocabulary size and coverage while reducing the open-vocabulary problem. Byte Pair Encoding (BPE), originally proposed as a compression algorithm, was later adapted to neural machine translation and became one of the most widely used subword methods. Other approaches, such as unigram language modeling, provide alternative segmentation strategies and highlight that tokenization itself is a modeling choice rather than a neutral preprocessing step.

Recent work has explored how pretrained tokenizers can be adapted to new domains. One line of research expands or augments the vocabulary with domain-specific tokens, aiming to improve coverage of specialized terms. Another line of work focuses on post-hoc refinement under a constrained vocabulary budget. AdaptBPE is particularly relevant to our study because it proposes replacing low-utility tokens with more useful alternatives while keeping the vocabulary size fixed and maintaining compatibility with pretrained models. Related studies have also examined trimming or removing problematic tokens, but the reported effects are mixed: vocabulary trimming may fail to improve performance consistently, while scaffold-token removal can improve token balance during training.

A further strand of work studies how tokenization affects downstream performance. Compression has often been treated as an intrinsic indicator of tokenizer quality, and some evidence suggests that better compression can correlate

with stronger downstream results. At the same time, more recent work argues that compression alone is insufficient and that tokenizer evaluation should consider multiple aspects of token behavior. Our work builds on this discussion by testing how a fixed-vocabulary replacement strategy behaves across models and classification tasks.

3 Methodology

Our method follows a fixed-vocabulary post-processing replacement strategy inspired by AdaptBPE. The goal is to improve token efficiency on a target corpus without changing the vocabulary size, retraining the tokenizer, or modifying model parameters. We first build a candidate list from the training corpus. Words are lowercased and counted by frequency. A word is kept as a candidate if it is split into more than one token by the original tokenizer, since only such words can potentially reduce sequence length after replacement. At the same time, we identify low-frequency token IDs from the original vocabulary, excluding special tokens. For a given replacement budget, we select a set of low-frequency token IDs to reclaim and a matched set of high-frequency candidate words to insert. This produces a one-to-one replacement table. Each inserted word is associated with its original multi-token sequence, while each reclaimed ID serves as the new single-token representation of that word. In this way, the vocabulary size remains unchanged. This design follows the core idea of lightweight tokenizer adaptation under a fixed vocabulary constraint. During tokenization, the input text is first encoded by the original tokenizer. We then apply a left-to-right greedy longest-match replacement over the token sequence. If a subsequence matches an entry in the replacement table, that subsequence is replaced with the corresponding reclaimed token ID. When multiple rules are applicable, the longest match is chosen. This prevents overlapping replacements and favors greater token reduction.

To measure compression, we compare the tokenized length before and after replacement. For each text, the reduction ratio is defined as the relative decrease in the number of tokens. We report the mean reduction ratio over a sample of training examples, following the common use of compression as an intrinsic indicator of tokenizer efficiency.

4 Experimental Setup

We evaluate the proposed tokenizer replacement strategy on two pretrained Transformer-based models, GPT-2 and RoBERTa. GPT-2 is used as a representative decoder-only architecture, while RoBERTa represents encoder-only pretrained models. This comparison allows us to test whether the same tokenizer adaptation strategy behaves differently across model types. Experiments are conducted on three English text classification datasets. IMDB is a large-scale binary sentiment classification dataset built from movie reviews. SST-2 is a sentence-level sentiment benchmark derived from the Stanford Sentiment Treebank. AG News is a four-class topic classification dataset constructed from news articles. These datasets were selected to provide variation in both task type and average input length.

For GPT-2 on IMDB, we evaluate replacement budgets of 0%, 1%, 3%, 5%, 7%, and 10% in order to trace a finer compression–accuracy curve. For RoBERTa, we evaluate 0%, 5%, and 10% on IMDB, SST-2, and AG News. In all cases, the original tokenizer with 0% replacement is used as the baseline.

To ensure fair comparison, all experiments use the same training configuration: maximum sequence length 128, batch size 8, learning rate $2e-5$, and 3 training epochs. Downstream performance is measured by classification accuracy, while tokenizer efficiency is measured by mean token reduction ratio. For the main settings, we report results across multiple random seeds and focus on both average performance and overall trend.

5 Results and Discussion

5.1 Main Results Overview

Table 1 summarizes the main results across all settings. Overall, the fixed-vocabulary replacement strategy does not behave uniformly across models and datasets. On GPT-2 with IMDB, token reduction increases with the replacement budget, while accuracy remains nearly unchanged. On RoBERTa with IMDB, similar compression gains are achieved, but accuracy decreases relative to the baseline. Across the RoBERTa experiments, SST-2 shows the largest compression gain together with the largest performance drop, whereas AG News is affected only slightly. These results suggest that the effectiveness of the AdaptBPE-style post-processing strategy depends strongly on both model architecture and task type.

Table 1 Main experimental results.

Model	Dataset	Budget	Accuracy ($\pm$ std)	Token Reduction Ratio
GPT-2	IMDB	0%	0.8895 ± 0.0010	0.00%
GPT-2	IMDB	5%	0.8883 ± 0.0013	1.56%
GPT-2	IMDB	10%	0.8891 ± 0.0015	2.00%
RoBERTa	IMDB	0%	0.9093 ± 0.0013	0.00%
RoBERTa	IMDB	5%	0.9014 ± 0.0036	1.60%
RoBERTa	IMDB	10%	0.9047 ± 0.0016	2.00%
RoBERTa	SST-2	0%	0.9343 ± 0.0037	0.00%
RoBERTa	SST-2	5%	0.9209 ± 0.0023	1.77%
RoBERTa	SST-2	10%	0.9079 ± 0.0013	2.54%
RoBERTa	AG News	0%	0.9519 ± 0.0011	0.00%
RoBERTa	AG News	5%	0.9494 ± 0.0009	0.41%
RoBERTa	AG News	10%	0.9486 ± 0.0004	0.56%

5.2 Cross-Architecture Comparison

The clearest architecture comparison comes from IMDB, where GPT-2 and RoBERTa are evaluated under similar replacement settings. GPT-2 remains stable across the tested budgets: accuracy changes only from 0.8895 at baseline to 0.8883 at 5% and 0.8891 at 10%, while compression reaches 1.56% and 2.00%. RoBERTa behaves differently under a comparable compression range. Its accuracy drops from 0.9093 at baseline to 0.9014 at 5% and 0.9047 at 10%. Thus, similar token reduction leads to much smaller performance changes in GPT-2 than in RoBERTa.

This result qualifies the plug-and-play view of tokenizer adaptation. AdaptBPE shows that post-hoc vocabulary refinement can improve compression without modifying model parameters, but our results indicate that such compatibility does not imply uniform robustness across architectures. In our setting, the decoder-only GPT-2 is clearly more stable than the encoder-only RoBERTa. A cautious interpretation is that RoBERTa may be more sensitive to perturbations in token sequence structure, although this remains a hypothesis rather than a demonstrated mechanism. This observation is consistent with prior findings that tokenizer adaptation does not guarantee uniform gains across different model settings.

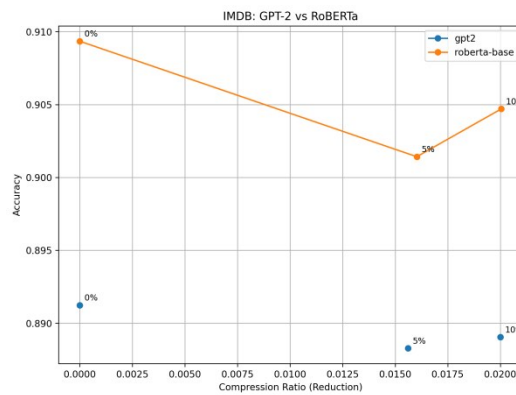


Figure 1 Accuracy-compression trade-off on IMDB for GPT-2 and RoBERTa

5.3 Cross-Dataset Trade-off

The RoBERTa experiments further show that the effect of replacement is highly task-dependent. SST-2 is the most sensitive setting: accuracy drops from 0.9343 to 0.9209 and then to 0.9079, while compression rises to 1.77% and 2.54%. AG News shows the opposite pattern: accuracy changes only from 0.9519 to 0.9494 and 0.9486, with compression remaining low at 0.41% and 0.56%. IMDB lies between these two extremes, with moderate compression and a smaller performance drop than SST-2.

An equally important observation is that compression alone does not determine downstream behavior. Although higher budgets generally increase token reduction, the corresponding accuracy change is not strictly monotonic. Similar inconsistencies have been observed in prior work on tokenizer evaluation. For example, on RoBERTa-IMDB, 10% replacement produces higher compression than 5% while also slightly recovering accuracy. This means that compression is informative, but not sufficient as a standalone indicator of task performance. This interpretation is consistent with prior work showing that compression is related to downstream performance, while also being more predictive in generation settings than in classification tasks. It is also aligned with more recent work arguing that tokenizer evaluation should not rely on compression alone. Our results provide another classification-based case where similar compression levels correspond to clearly different outcomes.

This pattern is also consistent with prior findings that token-level post-processing does not guarantee improvement and may sometimes lead to degradation.

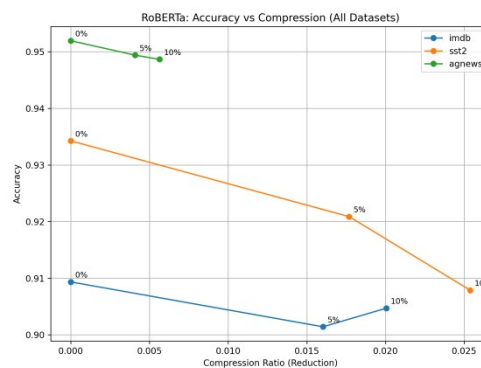


Figure 2 Accuracy-compression relationship across datasets for RoBERTa

5.4 Qualitative Error Analysis

Qualitative inspection helps explain why the same replacement strategy can help in some cases but hurt in others. In several SST-2 examples, the adapted tokenizer helps the model recover the correct label by making local negation cues easier to capture. In AG News, some improved predictions appear to be associated with clearer representation of domain-specific keywords. At the same time, the adapted tokenizer also introduces new errors, especially for short positive phrases or subtly expressed sentiment, where token merging may distort fine-grained semantic cues. Some cases remain difficult under both baseline and adapted settings, suggesting that they reflect ambiguity, irony, or conflicting sentiment rather than tokenization alone. Overall, the qualitative evidence suggests a trade-off between cue consolidation and semantic distortion, which helps explain why SST-2 is the most fragile setting in our experiments. Representative qualitative examples are provided in Appendix A.

6 Conclusion

This paper examined a fixed-vocabulary post-processing tokenizer replacement strategy across model architectures and text classification tasks. Using GPT-2 and RoBERTa on IMDB, SST-2, and AG News, we found that the method does not generalize uniformly across settings. On GPT-2, moderate token reduction can be achieved with little effect on classification accuracy. On RoBERTa, however, similar levels of compression are more likely to cause performance degradation. We also found strong task dependence: SST-2 is the most sensitive setting, while AG News is the most robust.

These results suggest that lightweight tokenizer adaptation is feasible, but its usefulness depends on both architecture and task properties. More importantly, compression alone is not sufficient to predict downstream behavior in classification settings. Future work could extend this analysis to larger models, generation tasks, and representation-level studies to better understand why some models are more sensitive to token replacement than others.

References

- [1] Sennrich, R., Haddow, B., Birch, A. (2016) Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin. pp. 1715–1725.
- [2] Gage, P. (1994) A new algorithm for data compression. *C Users J.*, 12: 23–38.
- [3] Kudo, T. (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. Melbourne. pp. 66–75.
- [4] Liyanage, V., Yvon, F. (2026) AdaptBPE: From general purpose to specialized tokenizers. *arXiv preprint*, arXiv:2601.21665.
- [5] Balde, G., Roy, S., Mondal, M., Ganguly, N. (2024) Adaptive BPE tokenization for enhanced vocabulary adaptation in finetuning pretrained language models. In: Findings of the 2024 Conference on Empirical Methods in Natural Language Processing. Miami. pp. 1–11.
- [6] Sachidananda, V., Kessler, J.S., Lai, Y.-A. (2021) Efficient domain adaptation of language models via adaptive tokenization. In: Proceedings of the Second Workshop on Domain Adaptation for NLP. Online. pp. 155–165.
- [7] Cognetta, M., Hiraoka, T., Okazaki, N., Sennrich, R., Pinter, Y. (2024) An analysis of BPE vocabulary trimming in neural machine translation. In: Proceedings of the Fifth Workshop on Insights from Negative Results in NLP. Mexico City. pp. 48–50.
- [8] Lian, H., Xiong, Y., Niu, J., Mo, S., Su, Z., Lin, Z., Chen, H., Liu, P., Han, J., Ding, G. (2024) Scaffold-BPE: Enhancing byte pair encoding for large language models with simple and effective scaffold token removal. *arXiv preprint*, arXiv:2404.17808.
- [9] Goldman, O., Caciularu, A., Eyal, M., Cao, K., Szpektor, I., Tsarfaty, R. (2024) Unpacking tokenization: Evaluating text compression and its correlation with model performance. *arXiv preprint*, arXiv:2403.06265.
- [10] Lotz, J.F., Lopes, A.V., Peitz, S., Setiawan, H., Emili, L. (2025) Beyond text compression: Evaluating tokenizers across scales. *arXiv preprint*, arXiv:2506.03101.
- [11] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019) Language models are unsupervised multitask learners. *OpenAI Technical Report*, 1: 1–24.
- [12] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V. (2019) RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint*, arXiv:1907.11692.
- [13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I. (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems* 30. Long Beach. pp. 5998–6008.
- [14] Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C. (2011) Learning word vectors for sentiment analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland. pp. 142–150.
- [15] Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C. (2013) Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle. pp. 1631–1642.
- [16] Zhang, X., Zhao, J., LeCun, Y. (2015) Character-level convolutional networks for text classification. In: *Advances in Neural Information Processing Systems* 28. Montreal. pp. 649–657.