Syllable Tokenization Does Not Improve Phonological Awareness in Large Language Models

Anonymous ACL submission

Abstract

Large Language Models often struggle with downstream task involving phonological awareness, despite increasing performance on natural language understanding benchmarks. One leading hypothesis for this discrepancy in performance is that tokenization along non-phonologically informed boundaries results in an inability to acquire phonological information from orthographic input, which makes up the majority of text-based Large Language Model training data. In this paper, we investigate this hypothesis by pretraining a Large Language Model on a Byte Pair Encoding tokenized corpus and an identical model on a syllable tokenized corpus. We compare their performance on syllable segmentation task, and a word segmentation task, but find no significant improvement from syllable tokenization on either task.

1 Introduction

Tokenization is the process through which large language models break down input text and convert it into integer representations. Currently, Byte Pair Encoding (BPE) is the most prevalent form of tokenization, utilized by OpenAI's GPT models, Deepseek, Eleuther's GPT-NeoX, among many others (Brown et al., 2020; DeepSeek-AI et al., 2025; Black et al., 2022) . BPE is a method of compression that recombines frequently occurring units into types; these types are indexed in a dictionary that is then used to segment text into corresponding ids (Figure 1; Gage 1994). Although there is active research on tokenizer-free models, most models still use a tokenizer (Clark et al., 2022; Pagnoni et al., 2024). This, in conjunction with the fact that tokenizer vocabularies are immutable once selected, means that the choice of tokenization is still an important part of LLM success.

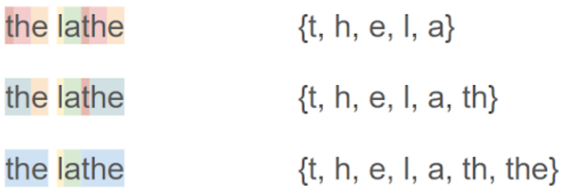

Figure 1: Character Level Byte Pair Encoding demonstration. The type vocabulary is on the right, starting with the initial dictionary and showcasing 2 merges

Despite BPE's success as a subword tokenization algorithm, it has been criticized for not aligning with recognizable morphological boundaries (Bostrom and Durrett, 2020) and scrutinized as the cause for LLMs subpar performance on phonological recognition tasks (Suvarna et al., 2024). Much prior work has attempted to address these issues; Bostrom and Durrett (2020) show that unigram language modeling (Kudo, 2018) produces types more closely aligned with morphological boundaries in English and Japanese.

In this work, we perform a comparative analysis of BPE against English orthographic syllable encoding, a tokenization strategy that uses phonologically informed syllable boundaries to break up words into subwords. We evaluate two equivalently trained GPT-2 sized models on NLI and Phonological Benchmarks, testing the hypothesis that phonologically informed tokenization strategies will improve downstream performance on tasks that require phonological awareness while preserving language understanding. This potential was raised by the paper PhonologyBench: Evaluating the Phonological Skills of Large Language Models by Suvarna et al. (2024) in their investigation of LLM phonological performance. Suvarna et al. (2024) benchmark the phonological abilities of several open and closed source large language models: Llama-2-13B-Chat, Mistal-7B-Instruct, Mixtral-8X7B-Instruct, GPT-3.5-Turbo, GPT-4, and Claude3-Sonnet. Despite these models' impressive sizes and performances on state of the art natural language evaluation tasks, Suvarna et al. demonstrate that these LLMs demonstrate poor understanding of the phonology of English; Claude-3-Sonnet performs the best on the sentence level syllable counting task, but still significantly below the human baseline (55.3% against 90%). Their tentative hypothesis is that the method of tokenization is to blame for the models' subpar performance. In response, we produce SYLGPT, a GPT-2 model trained on syllable segmented tokens.

Our contributions are:

- An open-source tool for syllabifying orthographic English words according to their phonetic boundaries[1].
- An analysis of orthographic syllable based tokenization in comparison to byte pair encoding on phonologically related tasks.

We find that there is no significant impact of syllable encoding on downstream performance on a syllable or word counting task.

2 Prior Work

Phonologically informed tokenization is not a purely novel idea; Mikolov et al. (2011) use naive syllable boundaries to tokenize words in a speech recognition task. Atuhurra et al. (2024) use syllable tokenization to improve translation accuracy for low resource languages. Velayuthan and Sarveswaran (2024) evaluate large language model performance in Tamil, Sinhala, and Hindi with grapheme pair encoding (GPE), finding better compression ratios with GPE than BPE.

In general, there is a push for more linguistically motivated tokenizers on the basis of improving performance on linguistic tasks and for improved model transparency. MorphPiece is a morphologically motivated tokenizer developed by Jabbar (2024). See their work for a more extensive review of the literature behind morphologically informed tokenization. Similar research has been conducted in a variety of languages outside of English, including Korean (Jeon et al., 2023) and Sanskrit (Sandhan et al., 2022).

3 Methods

3.1 Training Corpus

Byte Pair Encoding requires a tokenization training corpus to iterate over in order to build a vocabulary. We select the RadioTalk corpus as our model pretraining and tokenizer training corpus (Beeferman et al., 2019). RadioTalk was selected for the ease of pre tokenization and cleanliness; we choose to use NLTK word_tokenize as our pre tokenization method for both BPE and syllable encoding tokenizers (Loper and Bird, 2002). Pre tokenization is the step before tokenization where text is segmented into some simple-to-implement, presumably useful intermediary form like words at whitespace boundaries, before tokenization occurs with consideration to the pre tokenization boundaries (Mielke et al., 2021).

3.2 Syllable Definition

Suvarna et al. (2024) define a syllable as "a unit of pronunciation having one vowel sound, with or without surrounding consonants forming the whole or part of a word." There is one additional quality of syllables that is important to consider when transferring the archetypically phonological concept to the realm of orthography; the Maximal Onset Principle states that onsets should be as long as they can be (Kahn, 1976). This phonological principle does not typically align with the typical presentation of syllables in orthography. Some words are typically orthographically syllabified consistent to the maximal onset principle,e.g. "diploma" syllabified as "di.plo.ma" from the Merriam-Webster dictionary[2]. Other words are orthographically syllabified most consistently with morphological boundaries,e.g. "traumatic" syllabified as "trau.ma.tic" from the Merriam-Webster dictionary (which would be syllabified as "trau.ma.tic" if consistent with the maximal onset principle)[3]. It is for this reason that we opt to train our own syllabifier tool as a tokenizer rather than opt for a dictionary lookup. Moving forward, references to word syllabification will follow the Maximal Onset Principle, even if the resulting segmentation does not necessarily align with typical dictionary entries to preserve orthographic and phonological alignment.

3.3 Tokenizers

In order to draw a fair comparison between the syllable tokenizer and BPE tokenizer, we set the max vocabulary size of the BPE tokenizer based on the number of found types by the syllable tokenizer.

[1] The URL of a Github repository containing the code and data for this paper will appear here.

[2] `https://www.merriam-webster.com/dictionary/diploma`, retrieved on 7/25/2025.

[3] `https://www.merriam-webster.com/dictionary/traumatic`, retrieved on 7/25/2025.

Condition	Description
BPE Classifier	Byte-Pair Encoding with MLP output
Syllable Classifier	Syllable Encoding with MLP output
BPE Neuron	Byte-Pair Encoding with single neuron output
Syllable Neuron	Syllable Encoding with single neuron output

Table 1: Tested model descriptors and descriptions

The syllable tokenizer was trained on 1,000,000 lines of the RadioTalk corpus set aside for tokenizer training. The syllable tokenizer identified 17,019 types after a selection of numeral-related syllables were added (Appendix A).

The BPE tokenizer was trained on the same 1,000,000 lines of text, and merged to a vocabulary size of 17,019. The average type length in characters was 6.78. Merging was conducted in consideration of word and punctuation boundaries, with prepended spaces.

3.4 Models

We pre-train two parametrically identical GPT-2 sized models (Appendix A). To evaluate the base models on our tasks, we add and finetune two types of wrappers to both base models: a classifier and single neuron output wrapper. The classifier wrapper is a two layer network with a hidden layer of size 360 neurons and output linear layer size 31 trained on logit outputs from the second to last transformer block and a softmax cross entropy loss function, while the single unit output wrapper is a two layer network with a hidden layer of size 360 neurons and output size 1 trained on a mean square error loss function. The models with their wrappers will hereinafter be referred to by their tokenization paradigm and their wrapper (Table 1).

4 Task

4.1 Syllable Counting

We finetune each model with the wrapper on a sample of 794 sentences from the Syllable Counting task of PhonologyBench for 30 epochs (Suvarna et al., 2024). Performance after each epoch is evaluated on a validation set of 99 sentences. After full finetuning, test accuracy is evaluated on a set of 100 sentences.

4.2 Word Counting

We also evaluate both models on a word counting task to establish the degree to which the models are capable of learning morphological segments beyond syllables. A word counting task establishes a benchmark for the syllable model that extends performance beyond a potential simple token counting task. We use the same classifier and single neuron wrappers trained on the same objective and sentences as the syllable counting task, with target values set to the number of words in the sentence instead of the number of syllables. The finetuning procedure is identical to the finetuning procedure of the syllable counting task.

5 Results

We find that using syllable based tokenization does not always result in better performance on the syllable and wod counting tasks. However, we do find an interaction between the type of wrapper used and the resulting model performance on both the syllable and word counting tasks.

5.1 Syllable Counting

5.1.1 Training and Validation Loss

Training and Validation Losses for each model are presented in Figure 2. *BPE Neuron* achieves and maintains the lowest Train and Validation losses averaged across 30 runs with different seeds.

5.1.2 Test Accuracy

The *BPE Neuron* model achieves the highest average test accuracy score of 0.43 (Figure 3). This was significantly greater than the average test accuracy of the *Syllable Classifier* model (p<0.05, t=5.85, df=58), the *Syllable Neuron* model (p<0.05, t=7.10, df=58), and the *BPE Classifier* model (p<0.05, t=15.30, df=58). The BPE Classifier achieved the lowest average test accuracy score of 0.16, significantly lower than both the *Syllable Neuron* model (p<0.05, t=-7.73, df=58) and the *Syllable Classifier* model (p<0.05, t=-7.49, df=58).

5.2 Word Counting

Training and Validation Losses for each model are presented in Appendix B and C.

5.2.1 Test Accuracy

The *BPE Classifier* model achieves the highest test accuracy score of 0.47 (Figure 4), significantly greater than the *BPE Neuron* model (p<0.05,

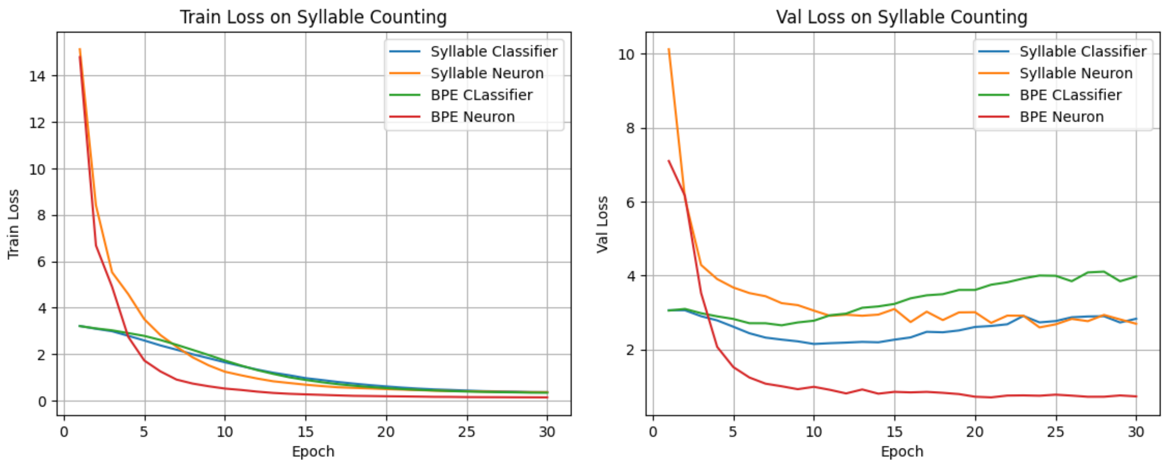

Figure 2: Training and validation loss across 30 epochs, averaged over 30 runs, for the syllable counting task.

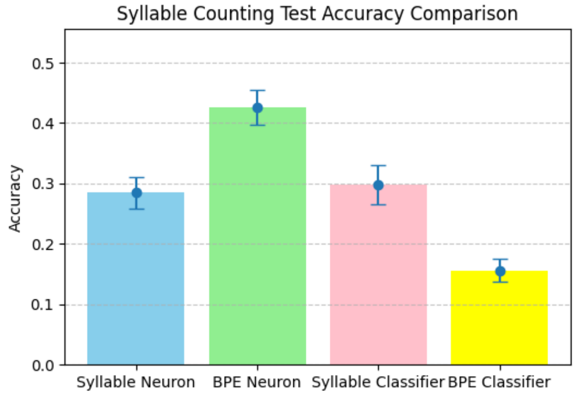

Figure 3: Test accuracy on the syllable counting task for each model averaged across 30 runs. Error bars display 95% CI.

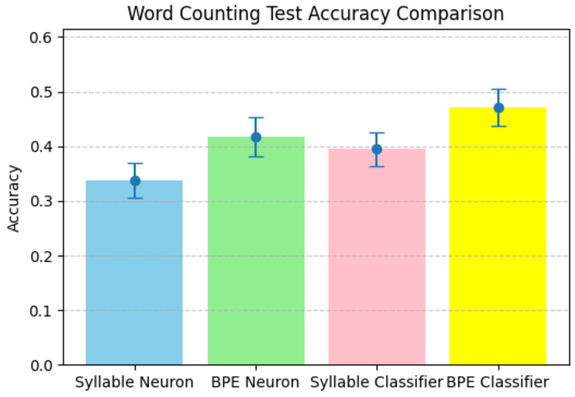

Figure 4: Test accuracy on the word counting task for each model averaged across 30 runs. Error bars display 95% CI.

t=2.12, df=58), the *Syllable Classifier* model (p<0.05, t=3.25, df=58), and the *Syllable Neuron* model p<0.05, t=5.51, df=58.

5.3 Cross task Performance

We also perform an analysis of each model's performance between tasks. The *Syllable Neuron* model performs significantly worse on the Syllable Counting task than the Word Counting task (p<0.05, t=-2.47, df=58). The *BPE Neuron* model does not perform significantly differently between the two tasks in a two sided t-test (p>0.05, t=0.39, df=58). The *Syllable Classifier* model does not perform significantly differently between the two tasks in a two sided t-test (p>0.05, t=-4.30, df=58). The textitBPE Classifier model performs significantly differently between the two tasks (p>0.05, t=-15.73, df=58), performing significantly better on the word counting task (p<0.05, t=15.73, df=58).

6 Discussion

Although we did not find that syllable tokenization unilaterally improved syllable recognition, we did find an interaction between wrapper types and tokenization paradigm that resulted in lower performance for BPE tokenization with the Classifier wrapper, and higher performance for BPE tokenization with the Neuron wrapper. Although these results do not necessarily show that syllable tokenization is more effective than BPE tokenization in preserving phonological information, they do demonstrate that the type of output wrapper used may contribute significantly to the results of finetuning. We did not find this same disparity in performance between the two types of syllable models trained on the syllable counting task, suggesting that the preservation of phonological information may be more stable with syllable tokenization.

7 Limitations

The broader generalizability of this study is limited; we examine a narrow definition of phonological tokenization (syllable based tokenization). Our evaluation set is also limited in size, reducing the power of our conclusions. In the future, we might consider expanding the task to a larger evaluation set, and investigation alternative phonologically-based tokenization paradigms, such as grapheme tokenization and morpho-phonological tokenization. Our study also only looks at English, a language with a deep orthographic structure. The results might be different for a language with shallower orthography like Spanish, or a language with logographic orthography such as Chinese.

References

Jesse Atuhurra, Hiroyuki Shindo, Hidetaka Kamigaito, and Taro Watanabe. 2024. Introducing Syllable Tokenization for Low-resource Languages: A Case Study with Swahili. *Preprint*, arXiv:2406.15358.

Doug Beeferman, William Brannon, and Deb Roy. 2019. RadioTalk: A large-scale corpus of talk radio transcripts. In *Interspeech 2019*, pages 564–568.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. *Preprint*, arXiv:2204.06745.

Kaj Bostrom and Greg Durrett. 2020. Byte Pair Encoding is Suboptimal for Language Model Pretraining. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4617–4624, Online. Association for Computational Linguistics.

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, and 12 others. 2020. Language Models are Few-Shot Learners. *Preprint*, arXiv:2005.14165.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. CANINE: Pre-training an Efficient Tokenization-Free Encoder for Language Representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *Preprint*, arXiv:2501.12948.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal archive*.

Haris Jabbar. 2024. MorphPiece : A Linguistic Tokenizer for Large Language Models. *Preprint*, arXiv:2307.07262.

Taehee Jeon, Bongseok Yang, Changhwan Kim, and Yoonseob Lim. 2023. Improving Korean NLP Tasks with Linguistically Informed Subword Tokenization and Sub-character Decomposition. *Preprint*, arXiv:2311.03928.

Daniel Kahn. 1976. *Syllable-Based Generalizations in English Phonology.* Thesis, Massachusetts Institute of Technology.

Andrej Karpathy. 2023. nanoGPT: The simplest, fastest repository for training/finetuning medium-sized GPTs. GitHub.

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.

Edward Loper and Steven Bird. 2002. NLTK: The Natural Language Toolkit. *arXiv preprint*.

Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. 2021. Between words and characters: A Brief History of Open-Vocabulary Modeling and Tokenization in NLP. *Preprint*, arXiv:2112.10508.

Tomasˇ Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan ˇCernocky. 2011. SUBWORD LANGUAGE MODELING WITH NEURAL NETWORKS.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srinivasan Iyer. 2024. Byte Latent Transformer: Patches Scale Better Than Tokens. *Preprint*, arXiv:2412.09871.

Jivnesh Sandhan, Rathin Singha, Narein Rao, Suvendu Samanta, Laxmidhar Behera, and Pawan Goyal. 2022. TransLIST: A Transformer-Based Linguistically Informed Sanskrit Tokenizer. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6902–6912, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. PhonologyBench: Evaluating Phonological Skills of Large Language Models. *Preprint*, arXiv:2404.02456.

Menan Velayuthan and Kengatharaiyer Sarveswaran. 2024. Egalitarian Language Representation in Language Models: It All Begins with Tokenizers. *Preprint*, arXiv:2409.11501.

A Model Parameters

Each model consists of 12 transformer block layers, 12 attention heads, embedding dimension of 768, and vocabulary size of 17,019. We use the AdamW optimizer with a learning rate of 6e-4, weight decay of 1e-1, beta1 and beta2 of 0.9, 0.95. Each model is trained for 4,000 iterations, training on 2,211,840 tokens per iteration. The total number of parameters is 98.03M. Hyperparameters and code are adapted from Karpathy (2023).

B Word Counting Training Loss Curves

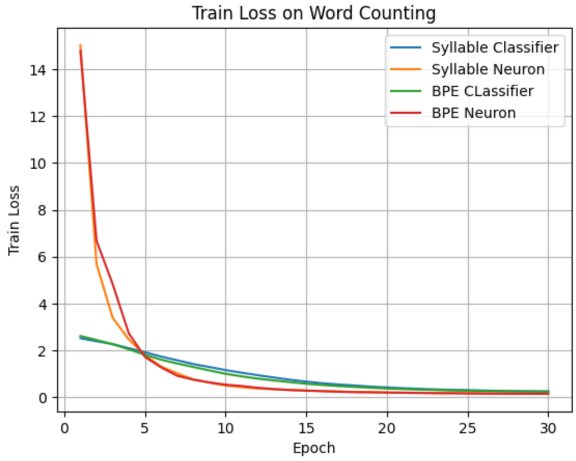

Figure 5: Training loss across 30 epochs, averaged over 30 runs, for the word counting task.

C Word Counting Validation Loss Curves

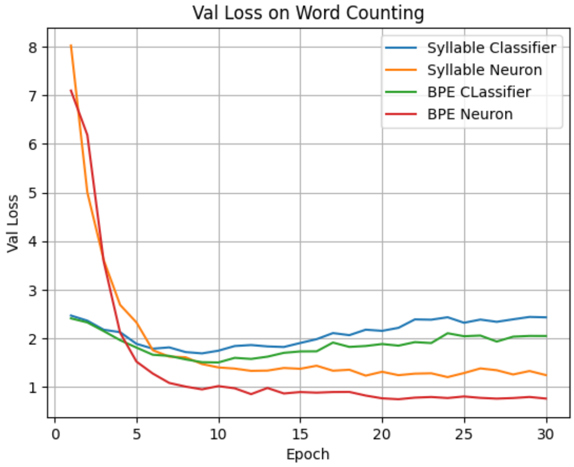

Figure 6: Validation loss across 30 epochs, averaged over 30 runs, for the word counting task.