Language Model Tokenizers Introduce Unfairness Between Languages

Aleksandar Petrov [1 2] **Emanuele La Malfa** [2] **Philip H.S. Torr** [1] **Adel Bibi** [1]

Abstract

Recent language models have shown impressive multilingual performance, even when not explicitly trained for it. Despite this, there are concerns about the quality of their outputs across different languages. In this paper, we show how disparity in the treatment of languages arises at the tokenization stage, well before a model is even invoked. The same text translated into different languages can have drastically different tokenization lengths, with differences up to 15 times in some cases. These disparities persist across the 17 tokenizers we evaluate, even if they are intentionally trained for multilingual support. Character-level and byte-level models also exhibit over 4 times the difference in the encoding length for some language pairs. This induces unfair treatment for some language communities in regard to the cost of accessing commercial language services, the processing time and latency, as well as the amount of content that can be provided as context to the models. Therefore, we make the case that we should train future language models using multilingually fair subword tokenizers.

1. Introduction

Language models are becoming increasingly important in natural language processing tasks, as they are capable of understanding and generating human-like language. As general-purpose technologies, it is also projected that Large Language Models (LLMs) will have a significant impact on the economy and the labour market (Teubner et al., 2023; Eloundou et al., 2023).

Such LLMs are often trained using large swaths of internet content regardless of language. Hence, these models often end up being multilingual, even if not by design. ChatGPT (OpenAI, 2022) is a prominent recent example (Bang et al., 2023; Jiao et al., 2023; Johnson, 2023). This is good: in line with the economic benefits of LLMs and LLM-derived technology, equal access is crucial, with multilingual support serving as a key component.

However, this multilingualism is treated as a curious emergent phenomenon rather than a carefully managed process. As LLM multilingualism emerges, we should pay attention to ensuring comparable performance and accessibility across the supported languages, regardless of whether by design or by chance.

This work demonstrates how the unequal treatment of languages arises at the tokenization stage, well before the language model sees any data at all. For instance, the ChatGPT (OpenAI, 2022) and GPT-4 (OpenAI, 2023) tokenizer uses about 1.6 times more tokens to encode the same text in Italian as it does in English, 2.6 times for Bulgarian and 3 times for Arabic. For Shan —the native language of people from the Shan State in Myanmar— the difference is 15 times. Unicode character and byte-level tokenization also result in drastically different encoding lengths across languages: byte-level representation of the same text is over 4 times longer for Burmese or Tibetan than Chinese.

We discuss three fairness implications of these differences in tokenization:

1. **Cost:** Commercial services charge users per token or Unicode character. In either case, these discrepancies lead to users of some languages paying at least 4 times more for the same task as users of English.
2. **Latency:** The number of tokens has a direct effect on the processing time for a task. Some languages can require twice the time to process the same content as English. This may be critical for

[1]Department of Engineering Science, University of Oxford, Oxford, UK [2]Department of Computer Science, University of Oxford, Oxford, UK. Correspondence to: Aleksandar Petrov <aleks@robots.ox.ac.uk>.

Workshop on Challenges in Deployable Generative AI at International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA. 2023. Copyright 2023 by the author(s).

real-time applications like emergency services.

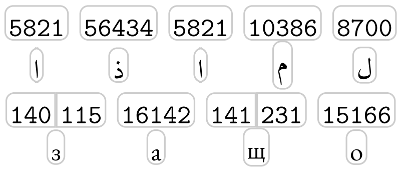
3. **Long-term dependency modelling:** Many models have a fixed-size context. Users of languages that are more token-efficient can use these systems to process or generate texts that may be more than an order of magnitude longer than users of other languages. This may lead to significant discrepancies in the quality of service.

Therefore, we make the case for *multilingual tokenization parity*, *i.e.*, that tokenizers should produce similar encoded lengths for the same content across languages. Hence, we advocate for multilingually fair tokenizers for the next generation of language models.

2. Measuring Tokenizer Parity

Using large corpora scraped from the internet results in *peculiar* choices for tokens. For instance, GPT-2 contains *glitch tokens* which can be usernames or concepts from games (Rumbelow & Watkins, 2023; Miles & Riley, 2023). As an example, `BuyableInstoreAndOnline`, likely coming from an online store backend, has a dedicated token. Another such token is `rawdownloadcloneembedreportprint`.

While such obscure terms get their own tokens, frequently used words in other languages require one token per letter, and in many cases, multiple tokens per letter. Take for example the Arabic "لماذا" and the Bulgarian "защо", both meaning "why":



The Arabic word is broken into letters and for Bulgarian some of the letters require two tokens to be represented, resulting in 6 tokens for a 4 letter word, all the while the English "why" has a dedicated token.

To demonstrate that the above examples are not anecdotal evidence, we introduce the notion of *tokenizer parity* to systematically assess how fairly a tokenizer treats equivalent sentences in different languages. Parity occurs when a tokenizer exhibits similar tokenized lengths for the same sentence in different languages. Take a sentence s_A in language A and its translation s_B to language B. Then, a tokenizer t achieves parity for A with respect to B at s_A and s_B if ${|t(s_A)|}/{|t(s_B)|} \approx 1$, where $t(s_A)$ is the tokenization of the sentence s_A and $|t(s_A)|$ represents its length. We refer to the ratio ${|t(s_A)|}/{|t(s_B)|}$ as the *premium* for A relative to B.[1]

Table 1: Premiums with respect to English on FLORES-200 for several **English-centric** models. The top and bottom three languages for any tokenizer and the ones discussed in the text are shown.

	GPT-2 RoBERTa	ChatGPT GPT-4	FlanT5
Bulgarian	5.51	2.64	—
Burmese	16.89	11.70	—
Chinese (Simplified)	3.21	1.91	—
Dzongkha	16.36	12.33	—
English	1.00	1.00	1.00
French	2.00	1.60	1.60
German	2.14	1.58	1.37
Italian	2.01	1.64	2.18
Japanese	3.00	2.30	—
Jingpho	2.65	2.35	3.41
Maori	2.45	2.35	3.28
Norwegian Bokmål	1.86	1.56	2.24
Odia	13.38	12.48	—
Pangasinan	1.66	1.57	2.18
Portuguese	1.94	1.48	2.21
Romanian	2.48	1.88	1.50
Santali	12.86	12.80	—
Shan	18.76	15.05	—
Spanish	1.99	1.55	2.23
Standard Arabic	4.40	3.04	—
Tumbuka	2.78	2.57	3.29
Vietnamese	4.54	2.45	—

3. Results

Languages vary significantly in how many tokens they require to encode the same content, as demonstrated in the examples in Section 2. Hence, we measure the tokenization premium of different tokenizers. To this end, we use the FLORES-200 parallel corpus, comprising of the same 2000 sentences taken from Wikipedia and human-translated to 200 different languages (Guzmán et al., 2019; Goyal et al., 2021; Costa-jussà et al., 2022).

3.1. Parity for English-centric Models

We consider several English-centric tokenizers: GPT-2 (Radford et al., 2019), RoBERTa (Liu et al., 2019), `cl100k_base` (the tokenizer used by ChatGPT and GPT-4 (OpenAI, 2022)), as well as FlanT5 (Chung et al., 2022). Some models, such as FlanT5, use a special `UNK` token to model unknown symbols not encountered during training. Hence, to ensure a fair comparison, we report only languages where no more than

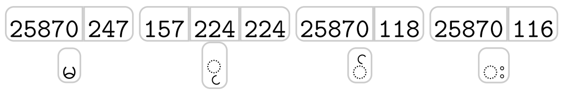
[1]The concurrent work by Ahia et al. (2023) also evaluates the tokenization premiums for different languages and reaches similar conclusions.

Table 2: Tokenizer premiums on the FLORES-200 dataset for **multilingual models**. The top and bottom two languages and the ones discussed in the text are shown. The premium is relative to English.

	XLM-R	M2M100	MBart50	mT5	BLOOM
Bulgarian	1.16	1.23	1.16	1.28	2.49
Chinese (Simp.)	0.97	1.05	0.97	0.92	0.95
Dzongkha	—	—	—	4.24	7.36
English	1.00	1.00	1.00	1.00	1.00
Indonesian	0.94	0.98	0.94	1.08	0.96
Italian	1.19	1.25	1.19	1.34	1.62
Japanese	1.11	1.20	1.11	0.90	1.81
Kabiyè	2.98	2.71	2.98	2.83	3.34
Santali	—	—	—	—	12.71
Shan	4.43	4.63	4.43	3.28	12.06
Std. Arabic	1.18	1.29	1.18	1.35	1.14
Std. Tibetan	—	—	—	3.68	6.66
Uyghur	1.41	3.00	1.41	2.57	3.67
Yue Chinese	0.93	1.03	0.93	0.95	0.93

10% of the input characters are mapped to UNK tokens. The results are presented in Table 1.

For GPT-2 and RoBERTa, Pangasinan, the cheapest language, is already 66% more expensive to process than English. ChatGPT and GPT-4 are slightly closer to parity, likely due to their larger vocabulary size. However, the cheapest languages, Portuguese, Pangasinan and German, still see a premium of 50% when compared to English. Shan has the worst tokenizer parity for all models with a premium of 15 for ChatGPT and GPT-4. Take as an example "မႂ်း", one of the Shan words for "you". The two models tokenize it as:



As the diacritics are encoded separately, there are four Unicode codepoints for this one Shan character, resulting in 9 tokens. In comparison, the English "you" has three characters but a single token.

FlanT5 has more than 10% UNK tokens for 42% of languages. It has a lower premium than the other tokenizers only for German and Romanian.

3.2. Parity for Multilingual Models

One would expect multilingual models to be close to tokenizer parity. We compare several multilingual models: XML-R (Conneau et al., 2020), M2M100 (Fan et al., 2021), MBart50 (Liu et al., 2020; Tang et al., 2020) and mT5 (Xue et al., 2020) and BLOOM (Scao et al., 2022) (see Table 2). Only BLOOM encodes all languages with no UNK tokens, thanks to its byte-level BPE tokenization. All models have languages with premiums of more than 4. Still, all models are closer to parity than the English-centric models in Section 3.1. However, none of the models uniformly reaches parity across all languages. Hence, even models which are intentionally designed to be multilingual suffer from a lack of tokenization parity.

Table 3: Tokenizer premiums on the FLORES-200 dataset for **byte-level models**. The top and bottom two languages and the ones discussed in the text are shown. The premium is relative to English.

	CANINE	ByT5
	UTF-32 bytes	UTF-8 bytes
Bulgarian	1.04	1.89
Burmese	1.24	3.51
Chinese (Simplified)	0.34	0.93
Chinese (Traditional)	0.32	0.89
Dzongkha	1.25	3.64
English	1.00	1.00
Italian	1.18	1.19
Japanese	0.44	1.27
Shan	1.42	3.94
Standard Arabic	0.88	1.60
Standard Tibetan	1.13	3.31
Tok Pisin	1.28	1.28
Tumbuka	1.30	1.32
Yue Chinese	0.31	0.87

3.3. Parity for Byte-level Tokenization Models

One can skip the tokenization altogether and work with characters or Unicode (*byte-level*) representations, allowing for full end-to-end training. Such models encode any character, even if unseen during training. CANINE (Clark et al., 2022) operates at the Unicode codepoint level. The CANINE tokenizer is thus equivalent to the UTF-32 encoding. ByT5 (Xue et al., 2022), on the other hand, uses the UTF-8 encoding. These models can represent any Unicode codepoint without an explicit tokenization step but there are still significant tokenization disparities.

For CANINE, Shan has a premium of 4.58 relative to Yue Chinese (Table 3). As CANINE provides a single token for each Unicode codepoint, Chinese is more token-efficient (with a premium range 0.31–0.34 relative to English for the three Chinese languages).

Tokenization disparity is also present in the ByT5 model. The tokenization premiums range from 0.87 (for Yue Chinese) to 3.94 (for Shan). The introduction of the variable-width UTF-8 encoding of Unicode characters in ByT5 creates another issue of unequal treatment. ASCII characters, which are sufficient for English, require only one byte. Other European scrips require two bytes, while Chinese, Japanese and Korean characters require three bytes. Therefore, the tokenization of Chinese and Japanese is about three times as long for ByT5 as it is for CANINE. Shan's premium of 3.94 is due to the fact that consonants and diacritics are separate codepoints, further exacerbating the disparity. The situation is similar for other languages like Dzongkha, Tibetan and Burmese.

4. Fairness Implications of Tokenization Length Differences

Our results show that no matter if one uses subword, multilingual, or byte-level tokenization, no the tokenizer gets close to parity. This leads to unfairness in the cost to access language models, the latency of the service and the amount of data that can be processed.

Cost. It is common to access LLMs as paid API services. When charging per token, the tokenization premiums discussed in Section 3 directly map to cost premiums. For example, ChatGPT and GPT-4 in Dzongkha, Odia, Santali or Shan cost more than 12 times more than in English. Another strategy is to charge per Unicode character. Still, as discussed in Section 3.3, the same content in different languages can have very different lengths when encoded in Unicode. Hence, both the per-token and the per-character strategies result in large disparities in the cost for users of different languages to use the same service.

Latency. High latency of real-time interactions can result in suboptimal experience and communication breakdowns. For customer support or emergency services, delays in response time can lead to miscommunication or delayed assistance. Our experiments with RoBERTa show a strong correlation between sequence length and execution time (Figure 1). As modern language models are partially autoregressive, longer outputs would take longer to generate. Therefore, tokenization disparities across languages also affect the latency and processing time for text in these languages. As expected, English has the shortest tokenization and one of the fastest processing times. Shan is on the other extreme with the longest tokenization length and execution time, almost twice that of English. Overall, the differences in processing time closely track the differences in tokenization length.

Long-term modelling. For transformers, inputs of greater length may impose a challenge to adequately reason over. Using ChatGPT and GPT-4, one can process less than a tenth of the content in languages like Burmese and Dzongkha than in English. Alongside inconveniencing the users of these languages, this can also result in diminished performance on automated systems like content moderation which are critical for hate speech prevention (Stecklow, 2018; Facebook, 2021). Hence, reduced long-term modelling capabilities for some languages could have severe real-world impacts.

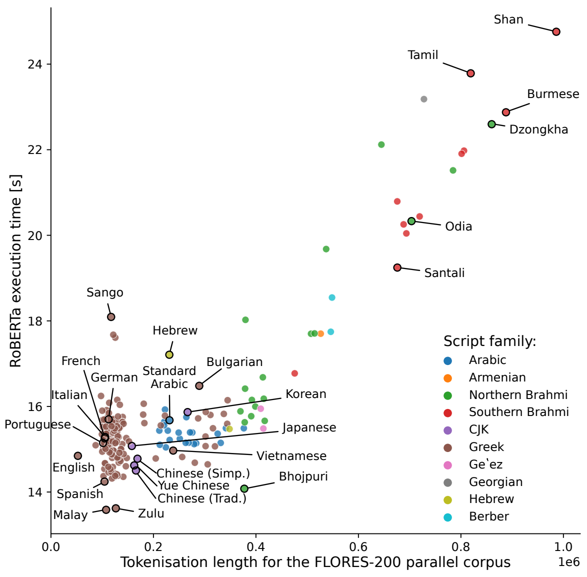


Figure 1: Processing time vs tokenization length. Each sentence is tokenized independently and the forward time of RoBERTa is averaged across 20 runs. The script families are only for illustration purposes.

5. Multilingual Tokenization Fairness

To achieve multilingual tokenization fairness, one needs a well-balanced and representative parallel corpus. Then, the tokenization procedure should start by encoding the input with one of the Unicode standards, as to be able to represent all characters from all languages. This would be then followed by building a subword tokenizer on top of it while ensuring parity across a set of languages. A subword tokenizer is necessary to handle the fact that different languages require different numbers of characters for the same content. The language model has to be trained using this fair tokenizer. This approach, combined with per-token pricing, will result in cost-parity, similar processing times across languages and similar effective context sizes.

6. Conclusion

This paper highlights the significant disparities in tokenization across languages which can lead to unequal treatment for certain language communities. Our findings reveal that even explicitly multilingual tokenizers exhibit differences in tokenization lengths of up to 13 times. Character-level and byte-level models also have encoding length discrepancies of over 4 times for some language pairs. These disparities have real-world implications including increased costs for accessing commercial language services, longer processing times and limitations on the amount of context available. To address these issues, we propose developing multilingually fair tokenizers. By achieving tokenization parity, we can promote fair access to language technologies across diverse linguistic communities.

Acknowledgements

We would like to thank Puyu Wang, Francisco Eiras, Ambre Bertrand and Carmen Scheidemann for their linguistic advice. We also extend special gratitude to Shinnosuke Takamichi and Hiroshi Saruwatari for open-sourcing the CPJD corpus for this project.

AB has received funding from the Amazon Research Awards. This work is supported by a UKRI grant Turing AI Fellowship (EP/W002981/1) and the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (EP/S024050/1). We also thank the Royal Academy of Engineering and FiveAI.

References

Ahia, O., Kumar, S., Gonen, H., Kasai, J., Mortensen, D. R., Smith, N. A., and Tsvetkov, Y. Do all languages cost the same? Tokenization in the era of commercial language models. *arXiv preprint arXiv:2305.13707*, 2023. URL https://arxiv.org/abs/2305.13707.

Bang, Y., Cahyawijaya, S., Lee, N., Dai, W., Su, D., Wilie, B., Lovenia, H., Ji, Z., Yu, T., Chung, W., Do, Q. V., Xu, Y., and Fung, P. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023. URL https://arxiv.org/abs/2302.04023.

Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *Advances in Neural Information Processing Systems*, 2000. URL https://proceedings.neurips.cc/paper/2000/hash/728f206c2a01bf572b5940d7d9a8fa4c-Abstract.html.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022. URL https://arxiv.org/abs/2210.11416.

Chung, J., Cho, K., and Bengio, Y. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016. URL https://aclanthology.org/P16-1160.

Clark, J. H., Garrette, D., Turc, I., and Wieting, J. Canine: Pre-training an Efficient Tokenization-Free Encoder for Language Representation. *Transactions of the Association for Computational Linguistics*, 2022. URL https://aclanthology.org/2022.tacl-1.5.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, É., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. In *Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://aclanthology.org/2020.acl-main.747.

Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022. URL https://arxiv.org/abs/2207.04672.

Eloundou, T., Manning, S., Mishkin, P., and Rock, D. GPTs are GPTs: An early look at the labor market impact potential of large language models. *arXiv preprint arXiv:2303.10130*, 2023. URL https://arxiv.org/abs/2303.10130.

Facebook. Sri Lanka human rights impact assessment, 2021. URL https://about.fb.com/wp-content/uploads/2021/03/FB-Response-Sri-Lanka-HRIA.pdf. Accessed on April 11, 2023.

Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., Baines, M., Celebi, O., Wenzek, G., Chaudhary, V., et al. Beyond English-centric multilingual machine translation. *The Journal of Machine Learning Research*, 2021. URL https://dl.acm.org/doi/abs/10.5555/3546258.3546365.

Gage, P. A new algorithm for data compression. *C Users Journal*, 1994.

Gao, Y., Nikolov, N. I., Hu, Y., and Hahnloser, R. H. Character-level translation with self-attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020. URL https://aclanthology.org/2020.acl-main.145.

Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., and Fan, A. The FLORES-101 evaluation benchmark for low-resource and multilingual machine translation. *Transactions of the Association for Computational Linguistics*, 2021. URL https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00474/110993/The-Flores-101-Evaluation-Benchmark-for-Low.

Guzmán, F., Chen, P.-J., Ott, M., Pino, J., Lample, G., Koehn, P., Chaudhary, V., and Ranzato, M. Two new evaluation datasets for low-resource machine translation: Nepali-English and Sinhala-English. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019. URL https://aclanthology.org/D19-1632.

Jiao, W., Wang, W., Huang, J., Wang, X., and Tu, Z. Is ChatGPT a good translator? Yes with GPT-4 as the engine. *arXiv preprint arXiv:2301.08745*, 2023. URL https://arxiv.org/abs/2301.08745.

Johnson. ChatGPT is a marvel of multilingualism. *The Economist*, 2023. URL https://www.economist.com/culture/2023/03/29/chatgpt-is-a-marvel-of-multilingualism.

Kudo, T. 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)*, 2018. URL https://aclanthology.org/P18-1007.

Kudo, T. and Richardson, J. 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*, 2018. URL https://aclanthology.org/D18-2012.

Lee, J., Cho, K., and Hofmann, T. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 2017. URL https://aclanthology.org/Q17-1026.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. URL https://arxiv.org/abs/1907.11692.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 2020. URL https://aclanthology.org/2020.tacl-1.47.

Mielke, S. J., Alyafeai, Z., Salesky, E., Raffel, C., Dey, M., Gallé, M., Raja, A., Si, C., Lee, W. Y., Sagot, B., et al. Between words and characters: A brief history of open-vocabulary modeling and tokenization in NLP. *arXiv preprint arXiv:2112.10508*, 2021. URL https://arxiv.org/abs/2112.10508.

Miles, R. and Riley, S. Glitch tokens – Computerphile, 2023. URL https://www.youtube.com/watch?v=WO2X3oZEJOA. Accessed on April 11, 2023.

OpenAI. Introducing ChatGPT, 2022. URL https://openai.com/blog/chatgpt. Accessed on April 11, 2023.

OpenAI. tiktoken, 2022. URL https://github.com/openai/tiktoken. Git commit: 82facf9.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. URL https://arxiv.org/abs/2303.08774.

Pfeiffer, J., Vulić, I., Gurevych, I., and Ruder, S. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://aclanthology.org/2021.emnlp-main.800.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Rumbelow, J. and Watkins, M. SolidGoldMagikarp (plus, prompt generation), 2023. URL https://www.alignmentforum.org/posts/aPeJE8bSo6rAFoLqg. Accessed on April 11, 2023.

Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagné, R., Luccioni, A. S., Yvon, F., Gallé, M., et al. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. URL https://arxiv.org/abs/2211.05100.

Schuster, M. and Nakajima, K. Japanese and Korean voice search. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012. URL https://ieeexplore.ieee.org/abstract/document/6289079.

Sennrich, R., Haddow, B., and Birch, A. 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)*, 2016. URL https://aclanthology.org/P16-1162.

Shao, Y., Hardmeier, C., and Nivre, J. Universal word segmentation: Implementation and interpretation. *Transactions of the Association for Computational Linguistics*, 2018. URL https://doi.org/10.1162/tacl_a_00033.

Stecklow, S. Hatebook. *Reuters*, 2018. URL https://www.reuters.com/investigates/special-report/myanmar-facebook-hate/. Accessed on April 11, 2023.

Sun, L., Hashimoto, K., Yin, W., Asai, A., Li, J., Yu, P., and Xiong, C. Adv-BERT: BERT is not robust on misspellings! Generating nature adversarial samples on BERT. *arXiv preprint arXiv:2003.04985*, 2020. URL https://arxiv.org/abs/2003.04985.

Tang, Y., Tran, C., Li, X., Chen, P.-J., Goyal, N., Chaudhary, V., Gu, J., and Fan, A. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*, 2020. URL https://arxiv.org/abs/2008.00401.

Teubner, T., Flath, C. M., Weinhardt, C., van der Aalst, W., and Hinz, O. Welcome to the era of ChatGPT et al: The prospects of large language models. *Business & Information Systems Engineering*, 2023. URL https://link.springer.com/article/10.1007/s12599-023-00795-x.

The Unicode Consortium. The Unicode standard, Version 15.0.0, 2022. URL https://www.unicode.org/versions/Unicode15.0.0.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Webster, J. J. and Kit, C. Tokenization as the initial phase in NLP. In *The International Conference on Computational Linguistics*, 1992. URL https://aclanthology.org/C92-4173.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. mT5: A massively multilingual pre-trained text-to-text transformer. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2020. URL https://aclanthology.org/2021.naacl-main.41.

Xue, L., Barua, A., Constant, N., Al-Rfou, R., Narang, S., Kale, M., Roberts, A., and Raffel, C. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 2022. URL https://aclanthology.org/2022.tacl-1.17.

A. Background on Tokenization

To enable the automatic processing of language, it must first be represented in a suitable form. The current practice is to use *tokenization* which is the process of turning natural language into sequences of *tokens* coming from a finite and pre-determined set called *vocabulary* (Webster & Kit, 1992). Each token is typically associated with an integer value. Language models process such sequences of integers, rather than sequences of characters or words. In this section, we offer a brief overview of the contemporary tokenization methods. For further details, we recommend the comprehensive survey by Mielke et al. (2021).

Word tokenization. The simplest tokenization method is splitting at white spaces, where each word is assigned its own token (Bengio et al., 2000). This approach, however, requires that all possible words are in the vocabulary which is not possible in practice. Therefore word tokenization often fails to handle cases like "won't", words spelled with accented characters like "naïve" or "açaí", speling mistakes and named entities like "Cottonshopeburnfoot" (Sun et al., 2020). This makes it unsuitable for representing *open vocabularies*, where the words encountered are not limited to a predetermined set. Furthermore, languages that do not use spaces to separate words, such as Chinese, Japanese and Burmese, pose additional challenges for this approach (Shao et al., 2018).

Subword tokenization. Hence, most current models use *subword tokenization*, where complex words are broken down into multiple tokens. Subword tokenization can efficiently handle complex terms by breaking them down into parts, *e.g.*, "Cottonshopeburnfoot" $\rightarrow$ "Cotton"+"shop"+"e"+"burn"+"foot". This approach can represent novel words, including misspelled ones, in an open vocabulary setting.

Subword vocabularies are usually data-based approaches which use large corpora to learn which subword sequences occur frequently in practice. Schuster & Nakajima (2012) introduced one of the first subword tokenizers, WordPiece, as a way to handle Japanese and Korean. Sennrich et al. (2016) proposed using Byte-Pair Encoding (BPE) (Gage, 1994) for learning subwords by merging the most frequently occurring pairs. BPE has since been widely used for most of the popular tokenizers. Kudo (2018) proposed an alternative approach via gradually pruning a large vocabulary. It removes tokens that are less likely to improve the performance of a simple unigram language model. Both methods rely on pre-tokenization (splitting on whitespaces, when available), which is not an invertible process. SentencePiece (Kudo & Richardson, 2018) addresses this de-tokenization ambiguity by treating whitespace as a special symbol, including it in the vocabulary, and supports both methods. SentencePiece with BPE is by far the most popular tokenization method for the models considered in this paper.

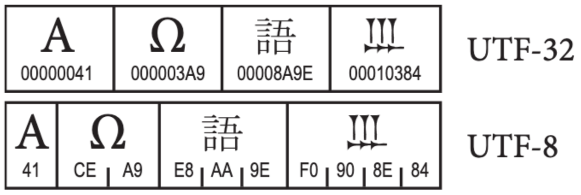


Figure 2: Comparison of variable width Unicode encoding (UTF-8) and fixed width encoding (UTF-32). Image adapted from (The Unicode Consortium, 2022).

Unicode support. Even if subword tokenization ensures that individual characters are in the vocabulary, this still leaves the question of which characters are to be included. A simple solution is to take the ASCII characters. However, this means that words in other scripts or accented letters will fall out of it. A common workaround is to represent strings outside the vocabulary as a special `UNK` token. However, if there are too many `UNK` tokens in an input, the performance of the model tends to deteriorate (Pfeiffer et al., 2021). Therefore, it is desirable that the number of `UNK` tokens in the input is kept as low as possible. A simple and commonly used solution is to base the vocabulary building on Unicode.

Unicode is a computing industry standard for representing text characters (The Unicode Consortium, 2022). Unicode supports virtually all languages (including many ancient ones, emojis and special characters) by assigning every grapheme, modifier, punctuation mark, control character or formatting character one of 1,114,112 integer *codepoints*. The codepoints can be represented in binary as the variable-width encoding UTF-8, which encodes every codepoint with one to four bytes, or the fixed-width UTF-32 which encodes all codepoints with four bytes (see Figure 2).

UTF-8 can therefore represent any string in any language as a string of bytes. As each byte can take only one out of 256 values, 256 tokens can be sufficient to encode all texts. In practice, this is usually combined with the BPE tokenizer. At first, the corpus is encoded as UTF-8 bytes and then BPE is run on top of it. As most characters occur frequently, BPE would assign them a dedicated token. If the model encounters a character that didn't exist in the training corpus (*e.g.*, the medium skin tone waving hand 👋🏽), it can still represent it byte-by-byte (`F0`+`9F`+`91`+`8B` for the waving hand and `F0`+`9F`+`8F`+`BD` for the skin tone modifier). This allows the vocabulary to efficiently represent frequently occurring words and rare characters. For example, the sentence "I love açaí" could be tokenized as "I "+"love "+"a"+`C3`+`A7`+"a"+`C3`+`AD`.

Byte-level and character-level tokenization. If we can represent any input with just 256 characters, then why bother with subword tokens? A key consideration is the sequence length. This is since transformers (Vaswani et al., 2017), the currently predominant deep learning architecture for language models, have attention layers with a quadratic complexity in the input length. Hence, as the number of characters is much longer than the sub-word tokenization, working on the character level has been traditionally considered computationally inefficient. However, Chung et al. (2016), Lee et al. (2017), Gao et al. (2020), Clark et al. (2022) and Xue et al. (2022) proposed various architectures working around this issue and operating directly on characters or UTF-8 bytes.

B. Extended Tables of Tokenization Premiums

Language	GPT-2	r50k_base	p50k_base	p50k_edit	cl100k_base	RoBERTa	GottBERT	CamemBERT	PhoBERT	RoCBert	XLM-RoBERTa
Acehnese (Arabic script)	4.78	4.78	4.78	4.78	3.78	4.78	6.67	—	—	—	1.94
Acehnese (Latin script)	2.16	2.16	2.16	2.16	1.98	2.16	2.10	1.85	1.64	2.84	1.57
Afrikaans	1.94	1.94	1.94	1.94	1.69	1.94	1.68	1.65	1.51	2.75	1.20
Akan	2.80	2.80	2.80	2.80	2.68	2.80	2.56	1.96	1.74	—	1.98
Amharic	7.79	7.79	7.79	7.79	7.68	7.79	6.99	—	—	—	1.34
Armenian	10.01	10.01	10.01	10.01	9.98	10.01	9.00	—	—	—	1.38
Assamese	9.79	9.79	9.78	9.78	6.20	9.79	11.22	—	—	—	1.90
Asturian	1.89	1.89	1.89	1.89	1.58	1.89	1.80	1.57	1.48	2.69	1.27
Awadhi	7.19	7.19	7.19	7.19	4.78	7.19	11.04	—	—	—	1.37
Ayacucho Quechua	2.20	2.20	2.20	2.20	2.08	2.20	2.17	1.84	1.68	2.95	1.59
Balinese	1.97	1.97	1.97	1.97	1.80	1.97	1.87	1.71	1.54	2.97	1.32
Bambara	2.66	2.66	2.66	2.66	2.57	2.66	2.48	1.84	1.68	—	1.82
Banjar (Arabic script)	5.03	5.03	5.03	5.03	3.80	5.03	7.46	—	—	—	1.92
Banjar (Latin script)	1.98	1.98	1.98	1.98	1.71	1.98	1.87	1.62	1.45	2.81	1.21
Bashkir	6.01	6.01	6.01	6.01	4.28	6.01	5.36	—	—	—	2.06
Basque	2.10	2.10	2.10	2.10	1.88	2.10	1.87	1.73	1.60	2.89	1.16
Belarusian	6.56	6.56	6.56	6.56	3.55	6.56	5.62	—	3.46	—	1.46
Bemba	2.46	2.46	2.46	2.46	2.23	2.46	2.28	2.01	1.84	3.28	1.76
Bengali	9.65	9.65	9.65	9.65	5.84	9.65	11.52	—	—	—	1.38
Bhojpuri	7.18	7.18	7.18	7.18	4.69	7.18	10.90	—	—	—	1.47
Bosnian	2.19	2.19	2.19	2.19	1.87	2.19	1.99	1.75	1.62	2.65	1.12
Buginese	2.20	2.20	2.20	2.20	1.98	2.20	2.01	1.73	1.62	2.85	1.51
Bulgarian	5.51	5.51	5.51	5.51	2.64	5.51	4.73	—	3.09	—	1.16
Burmese	16.89	16.89	16.89	16.89	11.70	16.89	15.19	—	—	—	1.72
Catalan	1.92	1.92	1.92	1.92	1.71	1.92	1.89	1.59	1.57	2.86	1.26
Cebuano	2.24	2.24	2.24	2.24	1.93	2.24	2.12	1.91	1.69	3.12	1.52
Central Atlas Tamazight	10.39	10.39	10.39	10.39	10.04	10.39	9.33	—	—	—	—
Central Aymara	2.32	2.32	2.32	2.32	2.17	2.32	2.18	1.94	1.76	2.84	1.70
Central Kanuri (Arabic script)	4.74	4.74	4.74	4.74	3.63	4.74	7.01	—	—	—	2.60
Central Kanuri (Latin script)	2.57	2.57	2.57	2.57	2.37	2.57	2.40	1.92	1.73	—	1.74
Central Kurdish	6.49	6.49	6.49	6.49	4.80	6.49	7.86	—	—	—	2.30
Chhattisgarhi	7.21	7.21	7.21	7.21	4.69	7.21	10.86	—	—	—	1.41
Chinese (Simplified)	3.21	3.21	3.21	3.21	1.91	3.21	3.95	—	—	1.00	0.97
Chinese (Traditional)	3.16	3.16	3.16	3.16	2.18	3.16	3.82	—	—	0.94	0.96
Chokwe	2.16	2.16	2.16	2.16	1.98	2.16	2.04	1.78	1.58	2.84	1.55
Crimean Tatar	2.49	2.49	2.49	2.49	2.12	2.49	2.25	2.01	1.85	—	1.38
Croatian	2.15	2.15	2.15	2.15	1.85	2.15	1.96	1.71	1.59	2.59	1.10
Czech	2.62	2.62	2.62	2.62	2.11	2.62	2.33	—	1.77	2.56	1.17
Danish	1.90	1.90	1.90	1.90	1.62	1.90	1.70	1.67	1.54	2.69	1.09
Dari	5.11	5.11	5.11	5.11	3.16	5.11	7.16	—	—	—	1.09
Dutch	1.97	1.97	1.97	1.97	1.59	1.97	1.73	1.68	1.58	2.92	1.14
Dyula	2.20	2.20	2.20	2.20	2.05	2.20	2.08	1.71	1.56	2.55	1.65
Dzongkha	16.36	16.36	16.36	16.36	12.33	16.36	16.12	—	—	—	—
Eastern Panjabi	7.90	7.90	7.90	7.90	7.87	7.90	11.43	—	—	—	1.57
Eastern Yiddish	6.63	6.63	6.63	6.63	5.57	6.63	8.55	—	—	—	1.58
Egyptian Arabic	4.23	4.23	4.23	4.23	2.96	4.23	6.88	—	—	—	1.17
English	1.00	1.00	1.00	1.00	1.00	1.00	1.35	1.20	1.20	2.60	1.00
Esperanto	2.03	2.03	2.03	2.03	1.87	2.03	1.85	1.62	1.52	2.61	1.20
Estonian	2.11	2.11	2.11	2.11	1.87	2.11	1.87	1.69	1.59	2.66	1.12
Ewe	2.90	2.90	2.90	2.90	2.75	2.90	2.66	2.02	1.75	—	2.01
Faroese	2.38	2.38	2.38	2.38	2.07	2.38	2.24	1.96	1.75	—	1.44
Fijian	2.30	2.30	2.30	2.30	2.15	2.30	2.26	1.82	1.67	2.94	1.72
Finnish	2.28	2.28	2.28	2.28	1.99	2.28	1.97	1.87	1.77	2.94	1.14
Fon	4.08	4.08	4.08	4.08	3.67	4.08	3.71	—	—	—	2.51
French	2.00	2.00	2.00	2.00	1.60	2.00	1.99	1.00	1.66	3.10	1.30
Friulian	2.07	2.07	2.07	2.07	1.85	2.07	1.98	1.66	1.59	2.79	1.56
Galician	1.91	1.91	1.91	1.91	1.56	1.91	1.87	1.63	1.56	2.88	1.13
Ganda	2.17	2.17	2.17	2.17	1.96	2.17	2.00	1.76	1.64	2.77	1.55
Georgian	13.85	13.85	13.85	13.85	9.85	13.85	12.43	—	—	—	1.34
German	2.14	2.14	2.14	2.14	1.58	2.14	1.00	1.85	1.67	3.12	1.17
Greek	6.54	6.54	6.54	6.54	5.15	6.54	6.73	—	3.73	3.00	1.45
Guarani	2.46	2.46	2.46	2.46	2.17	2.46	2.27	1.85	1.74	2.72	1.72
Gujarati	12.27	12.27	12.27	12.27	7.69	12.27	11.03	—	—	—	1.42
Haitian Creole	1.90	1.90	1.90	1.90	1.74	1.90	1.82	1.58	1.38	2.32	1.39
Halh Mongolian	6.42	6.42	6.42	6.42	3.77	6.42	5.71	—	3.26	—	1.21
Hausa	2.15	2.15	2.15	2.15	2.00	2.15	2.01	1.76	1.51	2.64	1.40
Hebrew	4.39	4.39	4.39	4.39	3.66	4.39	6.09	—	—	—	1.12
Hindi	7.46	7.46	7.46	7.46	4.79	7.46	11.25	—	—	—	1.25

Language	M2M100	MBart50	mT5	FlanT5	ByT5	CANINE	BLOOM	ArabicBERT	MuRIL	UTF-32	BERT Japanese
Acehnese (Arabic script)	1.89	1.94	1.79	—	1.51	0.85	2.65	—	—	0.85	—
Acehnese (Latin script)	1.47	1.57	1.44	2.55	1.09	1.07	1.74	2.63	2.02	1.07	2.11
Afrikaans	1.22	1.20	1.20	2.15	1.07	1.06	1.69	2.42	1.84	1.06	1.90
Akan	1.83	1.98	1.82	2.96	1.10	1.00	2.05	—	—	1.00	2.17
Amharic	1.42	1.34	1.73	—	1.72	0.67	5.07	—	—	0.67	—
Armenian	1.50	1.38	1.58	—	2.04	1.11	4.31	—	—	1.11	—
Assamese	2.24	1.90	1.94	—	2.54	0.96	1.41	—	1.24	0.96	—
Asturian	1.15	1.27	1.28	2.07	1.07	1.03	1.31	2.27	1.81	1.03	1.88
Awadhi	1.47	1.37	1.62	—	2.50	0.98	1.43	—	1.29	0.98	—
Ayacucho Quechua	1.54	1.59	1.42	2.59	1.08	1.07	1.83	2.68	1.95	1.07	2.12
Balinese	1.29	1.32	1.29	2.37	1.11	1.11	1.46	2.55	1.83	1.11	2.01
Bambara	1.72	1.82	1.65	2.70	1.04	0.96	1.89	—	—	0.96	2.00
Banjar (Arabic script)	1.93	1.92	1.76	—	1.69	0.93	2.47	1.90	—	0.93	—
Banjar (Latin script)	1.16	1.21	1.16	2.20	1.05	1.05	1.30	2.40	1.71	1.05	1.92
Bashkir	1.23	2.06	1.60	—	1.85	1.01	3.57	—	—	1.01	—
Basque	1.23	1.16	1.22	2.33	1.07	1.06	1.14	2.58	1.90	1.06	2.02
Belarusian	1.56	1.46	1.59	—	2.06	1.13	3.24	4.74	—	1.13	—
Bemba	1.67	1.76	1.57	3.01	1.23	1.23	1.92	3.01	2.17	1.23	2.44
Bengali	1.55	1.38	1.58	—	2.61	0.98	1.17	—	1.01	0.98	—
Bhojpuri	1.54	1.47	1.63	—	2.47	0.97	1.53	—	1.39	0.97	—
Bosnian	1.17	1.12	1.33	2.48	1.03	1.01	1.84	2.53	—	1.01	1.95
Buginese	1.49	1.51	1.44	2.51	1.09	1.06	1.71	2.64	1.96	1.06	2.07
Bulgarian	1.23	1.16	1.28	—	1.89	1.04	2.49	4.30	—	1.04	—
Burmese	2.21	1.72	1.56	—	3.51	1.24	10.05	—	—	1.24	—
Catalan	1.26	1.26	1.36	2.14	1.12	1.10	1.18	2.36	1.90	1.10	1.95
Cebuano	1.38	1.52	1.42	2.86	1.20	1.20	1.78	2.76	2.10	1.20	2.29
Central Atlas Tamazight	—	—	3.48	—	2.28	0.89	7.69	—	—	0.89	—
Central Aymara	1.64	1.70	1.57	2.71	1.07	1.05	1.94	2.63	1.98	1.05	2.16
Central Kanuri (Arabic script)	2.49	2.60	2.43	—	1.60	0.88	2.10	—	2.37	0.88	—
Central Kanuri (Latin script)	1.65	1.74	1.58	2.82	1.11	1.05	2.00	—	—	1.05	—
Central Kurdish	2.48	2.30	1.75	—	1.78	0.97	3.21	3.01	—	0.97	—
Chhattisgarhi	1.51	1.41	1.60	—	2.46	0.97	1.44	—	1.34	0.97	—
Chinese (Simplified)	1.05	0.97	0.92	—	0.93	0.34	0.95	—	—	0.34	0.82
Chinese (Traditional)	1.06	0.96	0.98	—	0.89	0.32	0.97	—	—	0.32	0.84
Chokwe	1.47	1.55	1.41	2.66	1.07	1.07	1.72	2.68	1.94	1.07	2.12
Crimean Tatar	1.37	1.38	1.32	2.80	1.13	1.03	2.07	2.64	—	1.03	—
Croatian	1.15	1.10	1.30	2.43	1.01	0.98	1.80	2.49	—	0.98	1.90
Czech	1.23	1.17	1.27	2.72	1.08	0.97	2.03	2.40	—	0.97	—
Danish	1.12	1.09	1.14	2.26	1.05	1.03	1.67	2.34	1.83	1.03	—
Dari	1.15	1.09	1.31	—	1.63	0.92	1.64	1.99	1.58	0.92	—
Dutch	1.18	1.14	1.17	2.19	1.11	1.11	1.71	2.52	1.91	1.11	1.98
Dyula	1.53	1.65	1.55	2.68	1.07	1.01	1.80	2.37	2.06	1.01	2.08
Dzongkha	—	—	4.24	—	3.64	1.25	7.36	—	—	1.25	—
Eastern Panjabi	1.68	1.57	2.11	—	2.59	1.01	1.43	—	1.35	1.01	—
Eastern Yiddish	1.61	1.58	1.66	—	1.94	1.08	4.42	4.39	—	1.08	—
Egyptian Arabic	1.27	1.17	1.28	—	1.56	0.86	1.16	1.05	1.89	0.86	—
English	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.83	1.00	1.00	1.49
Esperanto	1.38	1.20	1.19	2.19	1.02	1.00	1.65	2.26	—	1.00	—
Estonian	1.20	1.12	1.12	2.43	1.01	0.98	1.77	2.33	1.71	0.98	—
Ewe	1.86	2.01	1.82	2.85	1.07	0.97	2.11	—	—	0.97	—
Faroese	1.41	1.44	1.40	2.73	1.09	1.02	1.95	2.58	—	1.02	—
Fijian	1.62	1.72	1.59	3.02	1.17	1.17	1.99	3.01	2.01	1.17	2.29
Finnish	1.23	1.14	1.16	2.61	1.11	1.07	1.89	2.60	2.05	1.07	2.16
Fon	2.31	2.51	2.36	—	1.26	1.02	2.21	—	—	1.02	—
French	1.33	1.30	1.40	1.60	1.24	1.19	1.20	2.42	1.96	1.19	2.03
Friulian	1.47	1.56	1.52	2.30	1.13	1.10	1.70	2.33	1.94	1.10	1.92
Galician	1.14	1.13	1.31	2.18	1.13	1.11	1.27	2.37	1.91	1.11	1.97
Ganda	1.38	1.55	1.40	2.65	1.03	1.02	1.67	2.66	1.94	1.02	2.10
Georgian	1.56	1.34	1.55	—	2.95	1.10	4.98	—	—	1.10	—
German	1.24	1.17	1.19	1.37	1.18	1.17	1.68	2.63	2.02	1.17	2.04
Greek	1.58	1.45	1.65	—	2.17	1.20	3.81	4.93	—	1.20	—
Guarani	1.63	1.72	1.62	2.57	1.09	1.01	1.87	2.56	1.99	1.01	—
Gujarati	1.58	1.42	1.73	—	2.50	0.96	1.35	—	1.19	0.96	—
Haitian Creole	1.16	1.39	1.22	2.32	0.95	0.92	1.56	2.15	1.68	0.92	1.77
Halh Mongolian	1.34	1.21	1.48	—	1.91	1.04	3.38	—	—	1.04	—
Hausa	1.29	1.40	1.37	2.61	1.08	1.07	1.78	2.45	1.78	1.07	2.02
Hebrew	1.22	1.12	1.22	—	1.39	0.78	2.92	3.14	—	0.78	—
Hindi	1.36	1.25	1.59	—	2.55	1.00	1.28	—	1.16	1.00	—

Language	GPT-2	r50k_base	p50k_base	p50k_edit	cl100k_base	RoBERTa	GottBERT	CamemBERT	PhoBERT	RoCBert	XLM-RoBERTa
Hungarian	2.66	2.66	2.66	2.66	2.15	2.66	2.42	2.13	1.88	2.83	1.18
Icelandic	2.43	2.43	2.43	2.43	2.15	2.43	2.32	—	1.80	—	1.23
Igbo	3.42	3.42	3.42	3.42	2.44	3.42	3.14	2.12	1.78	2.58	2.12
Ilocano	2.26	2.26	2.26	2.26	2.05	2.26	2.14	1.92	1.69	3.15	1.61
Indonesian	1.98	1.98	1.98	1.98	1.55	1.98	1.85	1.67	1.50	2.90	0.94
Irish	2.56	2.56	2.56	2.56	2.33	2.56	2.37	2.10	1.86	2.99	1.50
Italian	2.01	2.01	2.01	2.01	1.64	2.01	1.93	1.63	1.60	3.10	1.19
Japanese	3.00	3.00	3.00	3.00	2.30	3.00	4.35	—	—	1.34	1.11
Javanese	1.93	1.93	1.93	1.93	1.73	1.93	1.83	1.66	1.45	2.76	1.15
Jingpho	2.65	2.65	2.65	2.65	2.35	2.65	2.55	2.13	1.84	3.12	1.94
Kabiyè	4.87	4.87	4.87	4.87	4.74	4.87	4.42	—	—	—	2.98
Kabuverdianu	1.93	1.93	1.93	1.93	1.72	1.93	1.78	1.55	1.45	2.55	1.35
Kabyle	2.50	2.50	2.50	2.50	2.47	2.50	2.35	1.90	1.71	2.35	1.84
Kamba	2.32	2.32	2.32	2.32	2.17	2.32	2.18	1.77	1.56	2.54	1.62
Kannada	13.69	13.69	13.68	13.68	8.90	13.69	12.50	—	—	—	1.36
Kashmiri (Arabic script)	6.19	6.19	6.19	6.19	4.62	6.19	7.60	—	—	—	1.93
Kashmiri (Devanagari script)	7.03	7.03	7.03	7.03	4.69	7.03	10.47	—	—	—	1.82
Kazakh	5.92	5.92	5.92	5.92	3.79	5.92	5.27	—	3.19	—	1.15
Khmer	15.33	15.33	15.33	15.33	8.88	15.33	13.78	—	—	—	1.62
Kikongo	2.17	2.17	2.17	2.17	1.99	2.17	2.17	1.73	1.65	2.90	1.58
Kikuyu	3.44	3.44	3.44	3.44	3.29	3.44	3.18	—	1.99	3.07	2.31
Kimbundu	2.33	2.33	2.33	2.33	2.13	2.33	2.22	1.89	1.72	2.90	1.64
Kinyarwanda	2.37	2.37	2.37	2.37	2.14	2.37	2.17	1.90	1.76	2.98	1.72
Korean	5.07	5.07	5.07	5.07	2.38	5.07	5.21	—	—	2.57	1.16
Kyrgyz	5.74	5.74	5.74	5.74	3.51	5.74	5.12	—	3.20	—	1.16
Lao	13.19	13.19	13.19	13.19	9.62	13.19	11.86	—	—	—	1.39
Latgalian	2.39	2.39	2.39	2.39	2.20	2.39	2.25	1.94	1.77	2.65	1.57
Ligurian	2.29	2.29	2.29	2.29	1.98	2.29	2.11	1.79	1.72	2.83	1.65
Limburgish	2.05	2.05	2.05	2.05	1.80	2.05	1.81	1.66	1.59	2.71	1.45
Lingala	2.03	2.03	2.03	2.03	1.86	2.03	1.98	1.63	1.51	2.80	1.52
Lithuanian	2.45	2.45	2.45	2.45	2.21	2.45	2.20	1.83	1.70	2.69	1.17
Lombard	2.37	2.37	2.37	2.37	2.04	2.37	2.13	1.82	1.69	2.70	1.71
Luba-Kasai	2.13	2.13	2.13	2.13	1.94	2.13	2.02	1.72	1.57	2.82	1.54
Luo	2.04	2.04	2.04	2.04	1.82	2.04	1.89	1.66	1.52	2.68	1.52
Luxembourgish	2.25	2.25	2.25	2.25	1.99	2.25	1.75	1.82	1.72	2.97	1.64
Macedonian	5.46	5.46	5.46	5.46	2.77	5.46	4.70	—	3.10	—	1.17
Magahi	7.22	7.22	7.22	7.22	4.70	7.22	10.88	—	—	—	1.41
Maithili	7.43	7.43	7.43	7.43	4.90	7.43	11.16	—	—	—	1.58
Malayalam	15.24	15.24	15.24	15.24	9.00	15.24	13.70	—	—	—	1.38
Maltese	2.69	2.69	2.69	2.69	2.41	2.69	2.43	2.06	1.88	2.68	1.96
Maori	2.45	2.45	2.45	2.45	2.35	2.45	2.38	2.02	1.76	2.74	1.86
Marathi	7.87	7.87	7.87	7.87	5.07	7.87	11.81	—	—	—	1.22
Meitei (Bengali script)	10.22	10.22	10.22	10.22	6.71	10.22	12.22	—	—	—	2.56
Mesopotamian Arabic	4.27	4.27	4.27	4.27	2.99	4.27	6.88	—	—	—	1.16
Minangkabau (Arabic script)	5.25	5.25	5.25	5.25	3.97	5.25	7.71	—	—	—	2.02
Minangkabau (Latin script)	1.97	1.97	1.97	1.97	1.77	1.97	1.89	1.67	1.50	2.84	1.31
Mizo	2.09	2.09	2.09	2.09	1.96	2.09	2.06	1.82	1.55	2.75	1.65
Moroccan Arabic	4.21	4.21	4.21	4.21	2.96	4.21	6.85	—	—	—	1.25
Mossi	2.54	2.54	2.54	2.54	2.32	2.54	2.35	1.81	1.66	2.20	1.78
Najdi Arabic	4.41	4.41	4.41	4.41	3.04	4.41	7.04	—	—	—	1.18
Nepali	7.59	7.59	7.59	7.59	4.79	7.59	11.29	—	—	—	1.13
Nigerian Fulfulde	1.99	1.99	1.99	1.99	1.85	1.99	1.85	1.54	1.39	2.23	1.46
North Azerbaijani	3.47	3.47	3.47	3.47	2.64	3.47	3.11	—	2.28	—	1.15
North Levantine Arabic	4.04	4.04	4.04	4.04	2.83	4.04	6.52	—	—	—	1.15
Northern Kurdish	2.45	2.45	2.45	2.45	2.20	2.45	2.24	1.98	1.68	2.56	1.38
Northern Sotho	2.32	2.32	2.32	2.32	2.18	2.32	2.19	1.89	1.77	2.90	1.75
Northern Uzbek	2.30	2.30	2.30	2.30	2.17	2.30	2.20	1.91	1.77	3.09	1.33
Norwegian Bokmål	1.86	1.86	1.86	1.86	1.56	1.86	1.65	1.64	1.52	2.63	1.07
Norwegian Nynorsk	1.93	1.93	1.93	1.93	1.64	1.93	1.69	1.68	1.55	2.64	1.17
Nuer	4.23	4.23	4.23	4.23	4.00	4.23	3.76	—	—	—	2.62
Nyanja	2.26	2.26	2.26	2.26	2.08	2.26	2.12	1.85	1.70	3.02	1.59
Occitan	2.07	2.07	2.07	2.07	1.83	2.07	1.99	1.68	1.66	2.97	1.50
Odia	13.38	13.38	13.38	13.38	12.48	13.38	12.06	—	—	—	1.45
Pangasinan	1.66	1.66	1.66	1.66	1.57	1.66	1.71	1.49	1.33	2.60	1.29
Papiamento	1.98	1.98	1.98	1.98	1.75	1.98	1.79	1.64	1.49	2.68	1.37
Plateau Malagasy	2.58	2.58	2.58	2.58	2.26	2.58	2.35	2.03	1.79	3.26	1.57
Polish	2.69	2.69	2.69	2.69	1.91	2.69	2.41	2.05	1.90	2.59	1.19
Portuguese	1.94	1.94	1.94	1.94	1.48	1.94	1.87	1.62	1.56	2.83	1.11

Language	M2M100	MBart50	mT5	FlanT5	ByT5	CANINE	BLOOM	ArabicBERT	MuRIL	UTF-32	BERT Japanese
Hungarian	1.28	1.18	1.26	2.99	1.16	1.05	2.07	2.56	2.31	1.05	—
Icelandic	1.29	1.23	1.32	2.81	1.09	0.99	1.99	2.45	—	0.99	—
Igbo	1.47	2.12	1.79	3.17	1.21	1.02	1.72	2.74	—	1.02	—
Ilocano	1.33	1.61	1.61	2.82	1.21	1.21	1.90	2.84	2.01	1.21	2.31
Indonesian	0.98	0.94	1.08	2.24	1.08	1.08	0.96	2.46	1.74	1.08	1.98
Irish	1.50	1.50	1.67	3.14	1.23	1.16	2.15	2.64	2.46	1.16	2.25
Italian	1.25	1.19	1.34	2.18	1.19	1.18	1.62	2.58	1.92	1.18	2.04
Japanese	1.20	1.11	0.90	—	1.27	0.44	1.81	1.85	—	0.44	1.00
Javanese	1.10	1.15	1.21	2.21	1.04	1.04	1.40	2.48	1.74	1.04	1.92
Jingpho	1.78	1.94	1.79	3.41	1.27	1.28	2.14	3.12	2.32	1.28	2.47
Kabiyè	2.71	2.98	2.83	—	1.37	1.09	3.34	—	—	1.09	—
Kabuverdianu	1.30	1.35	1.28	2.21	1.02	0.99	1.51	2.28	1.81	0.99	1.92
Kabyle	1.71	1.84	1.82	2.83	1.06	0.99	2.02	2.35	—	0.99	—
Kamba	1.52	1.62	1.52	2.69	1.01	0.98	1.77	2.42	—	0.98	—
Kannada	1.53	1.36	1.44	—	2.83	1.05	1.31	—	1.06	1.05	—
Kashmiri (Arabic script)	1.93	1.93	2.00	—	1.72	0.96	2.32	2.30	1.75	0.96	—
Kashmiri (Devanagari script)	1.86	1.82	1.79	—	2.40	0.96	1.85	—	1.75	0.96	—
Kazakh	1.28	1.15	1.20	—	1.89	1.03	3.23	—	—	1.03	—
Khmer	1.87	1.62	1.43	—	3.33	1.18	6.40	—	—	1.18	—
Kikongo	1.48	1.58	1.46	3.01	1.14	1.14	1.75	2.90	1.97	1.14	2.29
Kikuyu	2.17	2.31	2.18	—	1.30	1.17	2.48	2.86	—	1.17	—
Kimbundu	1.54	1.64	1.48	2.91	1.11	1.11	1.81	2.83	1.99	1.11	2.27
Kinyarwanda	1.63	1.72	1.51	2.76	1.13	1.11	1.58	2.81	2.15	1.11	2.24
Korean	1.21	1.16	1.27	—	1.20	0.51	2.79	2.37	—	0.51	—
Kyrgyz	1.66	1.16	1.32	—	1.88	1.02	3.02	—	—	1.02	—
Lao	1.61	1.39	1.27	—	2.73	0.99	8.70	—	—	0.99	—
Latgalian	1.51	1.57	1.46	2.70	1.05	0.99	1.99	2.48	—	0.99	—
Ligurian	1.59	1.65	1.69	2.54	1.17	1.10	1.81	2.52	2.05	1.10	—
Limburgish	1.38	1.45	1.38	2.25	1.07	1.04	1.75	2.41	1.92	1.04	1.90
Lingala	1.26	1.52	1.38	2.73	1.08	1.08	1.65	2.69	1.90	1.08	2.10
Lithuanian	1.25	1.17	1.23	2.58	1.06	1.00	1.94	2.43	—	1.00	—
Lombard	1.56	1.71	1.70	2.58	1.16	1.07	1.84	2.35	1.96	1.07	—
Luba-Kasai	1.43	1.54	1.37	2.48	1.08	1.08	1.68	2.63	1.89	1.08	2.11
Luo	1.43	1.52	1.41	2.55	1.05	1.05	1.68	2.47	1.87	1.05	2.02
Luxembourgish	1.32	1.64	1.46	2.24	1.15	1.12	1.89	2.56	2.17	1.12	1.96
Macedonian	1.24	1.17	1.29	—	1.89	1.04	2.50	—	—	1.04	—
Magahi	1.50	1.41	1.61	—	2.46	0.96	1.45	—	1.34	0.96	—
Maithili	1.64	1.58	1.74	—	2.53	0.98	1.56	—	1.50	0.98	—
Malayalam	1.59	1.38	1.35	—	3.10	1.13	1.38	—	1.18	1.13	—
Maltese	1.87	1.96	1.69	2.94	1.16	1.11	2.25	2.62	—	1.11	—
Maori	1.74	1.86	1.69	3.28	1.16	1.11	2.12	2.72	2.12	1.11	2.16
Marathi	1.38	1.22	1.52	—	2.67	1.01	1.21	—	1.06	1.01	—
Meitei (Bengali script)	2.59	2.56	2.21	—	2.77	1.03	2.35	—	2.34	1.03	—
Mesopotamian Arabic	1.27	1.16	1.28	—	1.56	0.86	1.15	1.01	1.93	0.86	—
Minangkabau (Arabic script)	1.99	2.02	1.84	—	1.74	0.96	2.58	2.05	—	0.96	—
Minangkabau (Latin script)	1.25	1.31	1.25	2.35	1.07	1.07	1.44	2.48	1.77	1.07	1.98
Mizo	1.54	1.65	1.57	2.76	1.10	1.10	1.83	2.61	1.92	1.10	2.05
Moroccan Arabic	1.33	1.25	1.29	—	1.56	0.86	1.26	1.14	1.91	0.86	—
Mossi	1.66	1.78	1.80	2.90	1.03	0.96	1.99	2.18	—	0.96	—
Najdi Arabic	1.30	1.18	1.35	—	1.60	0.88	1.15	1.00	1.97	0.88	—
Nepali	1.28	1.13	1.47	—	2.56	0.96	1.17	—	1.01	0.96	—
Nigerian Fulfulde	1.27	1.46	1.32	2.14	0.96	0.93	1.66	2.12	1.54	0.93	1.80
North Azerbaijani	1.26	1.15	1.35	—	1.26	1.09	2.30	3.17	—	1.09	—
North Levantine Arabic	1.24	1.15	1.23	—	1.48	0.82	1.13	1.00	1.83	0.82	—
Northern Kurdish	1.66	1.38	1.42	2.74	1.10	1.00	2.03	2.36	—	1.00	—
Northern Sotho	1.52	1.75	1.57	2.81	1.17	1.15	1.94	2.70	2.18	1.15	2.22
Northern Uzbek	1.37	1.33	1.38	2.80	1.13	1.13	1.98	2.89	2.12	1.13	2.29
Norwegian Bokmål	1.10	1.07	1.12	2.24	1.03	1.01	1.62	2.29	1.79	1.01	1.77
Norwegian Nynorsk	1.17	1.17	1.18	2.29	1.04	1.01	1.65	2.34	1.82	1.01	1.83
Nuer	2.44	2.62	2.42	—	1.32	1.08	2.79	—	—	1.08	—
Nyanja	1.55	1.59	1.35	2.71	1.12	1.12	1.78	2.78	2.02	1.12	2.15
Occitan	1.31	1.50	1.48	2.26	1.17	1.14	1.49	2.42	1.93	1.14	1.99
Odia	1.56	1.45	3.11	—	2.73	1.03	1.36	—	1.21	1.03	—
Pangasinan	1.23	1.29	1.22	2.18	1.00	1.00	1.45	2.27	1.54	1.00	1.80
Papiamento	1.32	1.37	1.36	2.28	1.08	1.05	1.54	2.28	1.80	1.05	1.94
Plateau Malagasy	1.49	1.57	1.59	3.00	1.26	1.22	2.07	3.00	2.33	1.22	2.37
Polish	1.26	1.19	1.31	2.82	1.13	1.06	2.14	2.78	—	1.06	—
Portuguese	1.14	1.11	1.29	2.21	1.12	1.09	1.12	2.37	1.88	1.09	1.85

Language	GPT-2	r50k_base	p50k_base	p50k_edit	cl100k_base	RoBERTa	GottBERT	CamemBERT	PhoBERT	RoCBert	XLM-RoBERTa
Romanian	2.48	2.48	2.48	2.48	1.88	2.48	2.28	1.84	1.75	2.94	1.24
Rundi	2.33	2.33	2.33	2.33	2.13	2.33	2.19	1.90	1.77	2.98	1.71
Russian	5.74	5.74	5.74	5.74	2.49	5.74	4.96	—	3.25	2.67	1.17
Samoan	2.57	2.57	2.57	2.57	2.29	2.57	2.28	1.95	1.79	2.83	1.92
Sango	2.23	2.23	2.23	2.23	2.08	2.23	2.08	1.80	1.59	2.64	1.66
Sanskrit	7.94	7.94	7.94	7.94	5.00	7.94	11.60	—	—	—	1.43
Santali	12.86	12.86	12.86	12.86	12.80	12.86	11.55	—	—	—	—
Sardinian	2.26	2.26	2.26	2.26	1.99	2.26	2.07	1.77	1.68	3.02	1.61
Scottish Gaelic	2.70	2.70	2.70	2.70	2.42	2.70	2.50	2.16	1.94	3.21	1.75
Serbian	5.34	5.34	5.34	5.34	2.92	5.34	4.61	—	2.94	—	1.18
Shan	18.76	18.76	18.76	18.76	15.05	18.76	16.88	—	—	—	4.43
Shona	2.29	2.29	2.29	2.29	2.13	2.29	2.14	1.89	1.73	3.07	1.63
Sicilian	2.27	2.27	2.27	2.27	2.01	2.27	2.12	1.71	1.64	2.75	1.58
Silesian	2.60	2.60	2.60	2.60	2.18	2.60	2.35	2.03	1.91	2.58	1.65
Sindhi	5.00	5.00	5.00	5.00	4.00	5.00	7.04	—	—	—	1.28
Sinhala	12.86	12.86	12.86	12.86	8.83	12.86	11.59	—	—	—	1.35
Slovak	2.52	2.52	2.52	2.52	2.14	2.52	2.23	1.91	1.75	2.66	1.18
Slovenian	2.11	2.11	2.11	2.11	1.88	2.11	1.96	1.73	1.59	2.63	1.13
Somali	2.36	2.36	2.36	2.36	2.18	2.36	2.23	2.02	1.78	3.01	1.39
South Azerbaijani	5.16	5.16	5.16	5.16	3.34	5.16	7.17	—	—	—	1.43
South Levantine Arabic	4.02	4.02	4.02	4.02	2.84	4.02	6.53	—	—	—	1.12
Southern Pashto	5.39	5.39	5.39	5.39	3.83	5.39	7.24	—	—	—	1.38
Southern Sotho	2.34	2.34	2.34	2.34	2.21	2.34	2.21	1.95	1.78	3.07	1.78
Southwestern Dinka	2.48	2.48	2.48	2.48	2.25	2.48	2.15	1.71	1.58	1.95	1.68
Spanish	1.99	1.99	1.99	1.99	1.55	1.99	1.95	1.72	1.64	3.09	1.20
Standard Arabic	4.40	4.40	4.40	4.40	3.04	4.40	7.03	—	—	—	1.18
Standard Arabic (Romanized)	2.51	2.51	2.51	2.51	2.45	2.51	2.38	2.06	1.86	3.10	1.94
Standard Latvian	2.54	2.54	2.54	2.54	2.35	2.54	2.37	2.01	1.87	2.74	1.23
Standard Malay	2.05	2.05	2.05	2.05	1.62	2.05	1.92	1.74	1.54	2.99	0.95
Standard Tibetan	14.93	14.93	14.93	14.93	11.27	14.93	14.66	—	—	—	—
Sundanese	2.02	2.02	2.02	2.02	1.82	2.02	1.88	1.66	1.49	2.77	1.22
Swahili	2.13	2.13	2.13	2.13	1.95	2.13	2.02	1.70	1.58	2.76	1.16
Swati	2.31	2.31	2.31	2.31	2.16	2.31	2.15	1.92	1.74	3.13	1.61
Swedish	1.95	1.95	1.95	1.95	1.58	1.95	1.64	1.69	1.58	2.65	1.07
Tagalog	2.28	2.28	2.28	2.28	2.06	2.28	2.20	2.00	1.74	3.28	1.43
Tajik	6.09	6.09	6.09	6.09	3.64	6.09	5.40	—	3.39	—	2.14
Tamasheq (Latin script)	2.39	2.39	2.39	2.39	2.22	2.39	2.18	1.79	1.55	—	1.71
Tamasheq (Tifinagh script)	10.43	10.43	10.43	10.43	10.13	10.43	9.38	—	—	—	—
Tamil	15.58	15.58	15.58	15.58	7.65	15.58	14.01	—	—	—	1.35
Tatar	5.82	5.82	5.82	5.82	3.75	5.82	5.18	—	—	—	1.81
Ta'izzi-Adeni Arabic	4.34	4.34	4.34	4.34	3.01	4.34	6.96	—	—	—	1.17
Telugu	13.09	13.09	13.09	13.09	8.34	13.09	11.77	—	—	—	1.33
Thai	9.05	9.05	9.05	9.05	4.39	9.05	8.89	—	3.39	—	1.08
Tigrinya	7.88	7.88	7.88	7.88	7.80	7.88	7.09	—	—	—	1.97
Tok Pisin	2.21	2.21	2.21	2.21	2.04	2.21	2.09	1.99	1.74	3.23	1.73
Tosk Albanian	2.65	2.65	2.65	2.65	2.25	2.65	2.39	2.17	2.02	2.90	1.32
Tsonga	2.45	2.45	2.45	2.45	2.26	2.45	2.29	2.03	1.76	3.09	1.79
Tswana	2.39	2.39	2.39	2.39	2.28	2.39	2.27	2.00	1.86	3.14	1.85
Tumbuka	2.78	2.78	2.78	2.78	2.57	2.78	2.61	2.21	2.00	3.49	1.92
Tunisian Arabic	4.20	4.20	4.20	4.20	2.93	4.20	6.78	—	—	—	1.20
Turkish	2.43	2.43	2.43	2.43	1.91	2.43	2.17	1.97	1.81	—	1.04
Turkmen	2.82	2.82	2.82	2.82	2.40	2.82	2.37	2.13	1.95	2.88	1.78
Twi	2.62	2.62	2.62	2.62	2.51	2.62	2.42	1.88	1.66	—	1.88
Ukrainian	5.75	5.75	5.75	5.75	3.00	5.75	4.98	—	3.10	—	1.21
Umbundu	2.24	2.24	2.24	2.24	2.01	2.24	2.07	1.78	1.63	2.73	1.57
Urdu	6.30	6.30	6.30	6.30	4.39	6.30	7.74	—	—	—	1.23
Uyghur	7.16	7.16	7.16	7.16	5.19	7.16	8.68	—	—	—	1.41
Venetian	2.00	2.00	2.00	2.00	1.70	2.00	1.86	1.60	1.47	—	1.36
Vietnamese	4.54	4.54	4.54	4.54	2.45	4.54	4.12	—	1.00	2.55	1.18
Waray	2.38	2.38	2.38	2.38	1.95	2.38	2.17	1.99	1.71	3.24	1.55
Welsh	2.34	2.34	2.34	2.34	2.12	2.34	2.24	2.01	1.83	2.76	1.43
West Central Oromo	2.53	2.53	2.53	2.53	2.32	2.53	2.32	2.07	1.93	3.23	1.78
Western Persian	5.32	5.32	5.32	5.32	3.28	5.32	7.38	—	—	—	1.10
Wolof	2.14	2.14	2.14	2.14	1.92	2.14	2.01	1.71	1.54	2.41	1.60
Xhosa	2.26	2.26	2.26	2.26	2.06	2.26	2.11	1.88	1.68	2.93	1.50
Yoruba	3.89	3.89	3.89	3.89	2.96	3.89	3.55	—	2.00	2.28	2.27
Yue Chinese	3.09	3.09	3.09	3.09	2.12	3.09	3.75	—	—	0.92	0.93
Zulu	2.41	2.41	2.41	2.41	2.20	2.41	2.23	1.96	1.77	3.12	1.55

Language	M2M100	MBart50	mT5	FlanT5	ByT5	CANINE	BLOOM	ArabicBERT	MuRIL	UTF-32	BERT Japanese
Romanian	1.29	1.24	1.37	1.50	1.19	1.13	1.91	2.44	—	1.13	—
Rundi	1.63	1.71	1.52	2.78	1.12	1.12	1.64	2.81	2.13	1.12	2.24
Russian	1.22	1.17	1.27	—	1.98	1.09	2.48	4.57	—	1.09	—
Samoan	1.80	1.92	1.92	3.09	1.22	1.16	2.13	2.86	2.22	1.16	2.31
Sango	1.53	1.66	1.63	3.14	1.12	1.09	1.80	2.64	2.05	1.09	2.23
Sanskrit	1.69	1.43	1.65	—	2.63	0.98	1.63	—	1.21	0.98	—
Santali	—	—	—	—	2.79	1.06	12.71	—	—	1.06	—
Sardinian	1.51	1.61	1.57	2.46	1.19	1.16	1.73	2.52	1.98	1.16	2.04
Scottish Gaelic	1.61	1.75	1.85	3.24	1.28	1.24	2.25	2.86	2.27	1.24	2.22
Serbian	1.26	1.18	1.30	—	1.80	0.99	2.57	—	—	0.99	—
Shan	4.63	4.43	3.28	—	3.94	1.42	12.06	—	—	1.42	—
Shona	1.58	1.63	1.35	2.79	1.12	1.12	1.80	2.83	2.06	1.12	2.21
Sicilian	1.53	1.58	1.53	2.46	1.11	1.05	1.80	2.58	1.84	1.05	—
Silesian	1.59	1.65	1.57	2.87	1.10	1.04	2.16	2.77	—	1.04	—
Sindhi	1.30	1.28	1.74	—	1.60	0.91	2.51	—	1.22	0.91	—
Sinhala	1.53	1.35	1.66	—	2.64	1.00	8.21	—	—	1.00	—
Slovak	1.24	1.18	1.30	2.74	1.09	1.00	2.01	2.47	—	1.00	—
Slovenian	1.19	1.13	1.20	2.42	1.02	1.00	1.81	2.50	—	1.00	1.94
Somali	1.37	1.39	1.48	3.06	1.14	1.14	2.03	2.78	2.05	1.14	2.27
South Azerbaijani	1.50	1.43	1.42	—	1.63	0.89	1.81	2.03	1.72	0.89	—
South Levantine Arabic	1.22	1.12	1.24	—	1.49	0.83	1.12	1.01	1.82	0.83	—
Southern Pashto	1.40	1.38	1.64	—	1.66	0.95	2.55	—	—	0.95	—
Southern Sotho	1.60	1.78	1.59	2.92	1.21	1.20	1.96	2.93	2.16	1.20	2.30
Southwestern Dinka	1.55	1.68	1.58	—	0.96	0.86	1.82	—	—	0.86	—
Spanish	1.21	1.20	1.31	2.23	1.21	1.19	1.21	2.51	1.98	1.19	2.10
Standard Arabic	1.29	1.18	1.35	—	1.60	0.88	1.14	1.00	1.97	0.88	—
Standard Arabic (Romanized)	1.83	1.94	1.73	2.94	1.17	1.17	2.15	2.92	2.28	1.17	2.45
Standard Latvian	1.29	1.23	1.30	2.78	1.11	1.02	2.08	2.47	—	1.02	—
Standard Malay	1.00	0.95	1.11	2.32	1.12	1.11	1.07	2.53	1.80	1.11	2.03
Standard Tibetan	—	—	3.68	—	3.31	1.13	6.66	—	—	1.13	—
Sundanese	1.10	1.22	1.22	2.32	1.05	1.04	1.48	2.43	1.80	1.04	1.95
Swahili	1.20	1.16	1.25	2.66	1.05	1.05	1.24	2.64	1.86	1.05	2.13
Swati	1.44	1.61	1.41	2.80	1.12	1.13	1.83	2.84	2.09	1.13	2.26
Swedish	1.10	1.07	1.11	2.22	1.04	1.01	1.65	2.20	1.90	1.01	1.79
Tagalog	1.43	1.43	1.46	2.85	1.26	1.26	1.85	2.84	2.08	1.26	2.39
Tajik	2.06	2.14	1.62	—	2.01	1.11	3.29	4.36	—	1.11	—
Tamasheq (Latin script)	1.57	1.71	1.64	2.55	1.01	0.95	1.90	—	—	0.95	—
Tamasheq (Tifinagh script)	—	—	3.59	—	2.29	0.94	7.74	—	—	0.94	—
Tamil	1.55	1.35	1.26	—	3.17	1.17	1.27	—	1.06	1.17	—
Tatar	1.54	1.81	1.41	—	1.85	1.01	3.15	—	—	1.01	—
Ta'izzi-Adeni Arabic	1.28	1.17	1.32	—	1.58	0.87	1.15	1.01	1.94	0.87	—
Telugu	—	1.33	1.42	—	2.68	1.01	1.33	—	1.21	1.01	—
Thai	1.27	1.08	0.99	—	2.75	0.96	4.63	—	—	0.96	—
Tigrinya	1.91	1.97	2.03	—	1.75	0.69	5.16	—	—	0.69	—
Tok Pisin	1.65	1.73	1.65	2.76	1.28	1.28	1.92	2.94	2.10	1.28	2.35
Tosk Albanian	1.36	1.32	1.48	3.09	1.20	1.12	2.17	2.66	2.52	1.12	—
Tsonga	1.69	1.79	1.61	3.13	1.20	1.20	2.01	3.01	2.19	1.20	2.46
Tswana	1.68	1.85	1.68	3.01	1.25	1.25	2.02	2.95	2.25	1.25	2.35
Tumbuka	1.88	1.92	1.61	3.29	1.32	1.30	2.19	3.27	—	1.30	—
Tunisian Arabic	1.29	1.20	1.29	—	1.54	0.85	1.19	1.04	1.90	0.85	—
Turkish	1.15	1.04	1.12	2.67	1.12	1.03	1.96	2.65	—	1.03	—
Turkmen	1.71	1.78	1.68	2.87	1.17	1.06	2.19	2.63	—	1.06	—
Twi	1.74	1.88	1.71	2.85	1.05	0.98	1.81	—	—	0.98	2.08
Ukrainian	1.28	1.21	1.33	—	1.86	1.02	2.75	4.28	—	1.02	—
Umbundu	1.49	1.57	1.47	2.72	1.05	1.01	1.74	2.66	1.94	1.01	1.99
Urdu	1.30	1.23	1.52	—	1.76	0.99	1.36	2.65	1.26	0.99	—
Uyghur	3.00	1.41	2.57	—	1.97	1.07	3.67	—	—	1.07	—
Venetian	1.31	1.36	1.36	2.21	1.06	1.01	1.57	2.27	1.84	1.01	1.84
Vietnamese	1.15	1.18	1.95	—	1.39	1.05	1.27	2.52	—	1.05	—
Waray	1.45	1.55	1.45	2.66	1.25	1.25	1.80	2.91	2.15	1.25	2.26
Welsh	1.44	1.43	1.70	3.12	1.07	1.07	2.09	2.83	2.32	1.07	2.20
West Central Oromo	1.49	1.78	1.69	3.16	1.20	1.19	2.19	2.98	2.17	1.19	2.44
Western Persian	1.17	1.10	1.34	—	1.70	0.94	1.78	2.03	1.62	0.94	—
Wolof	1.40	1.60	1.44	2.62	1.00	0.96	1.68	2.34	1.93	0.96	1.88
Xhosa	1.37	1.50	1.35	2.73	1.06	1.06	1.67	2.78	2.05	1.06	2.17
Yoruba	1.74	2.27	2.06	—	1.28	0.97	1.64	2.26	—	0.97	—
Yue Chinese	1.03	0.93	0.95	—	0.87	0.31	0.93	—	—	0.31	0.82
Zulu	1.35	1.55	1.40	2.84	1.12	1.12	1.76	2.96	2.15	1.12	2.29