A Multi-dimensional Evaluation of Tokenizer-free Multilingual Pretrained Models

Jimin Sun[1,2] **Patrick Fernandes**[1] **Xinyi Wang**[1] **Graham Neubig**[1]
[1]Language Technologies Institute, Carnegie Mellon University [2]Kakao Enterprise
{jimins2,pfernand,xinyiw1,gneubig}@cs.cmu.edu

Abstract

Recent works on tokenizer-free multilingual pretrained models show promising results in improving cross-lingual transfer and reducing engineering overhead compared to subword-based alternatives. However, previous work mainly focuses on reporting accuracy on a limited set of tasks and data settings, placing less emphasis on other important factors when tuning and deploying the models in practice, such as memory usage, inference speed, and finetuning data efficiency. We attempt to fill this gap by performing a comprehensive empirical comparison of multilingual tokenizer-free and subword-based models considering the various dimensions. Surprisingly, we find that subword-based models might still be the most practical choice in many settings, achieving better performance for lower inference latency and memory usage. Based on these results, we encourage future work in tokenizer-free methods to consider these factors when designing and evaluating new models.[1]

1 Introduction

Several recent results (Clark et al., 2022; Xue et al., 2022) have excited the research community with the possibility of "tokenizer-free" models, character-level and byte-level models, as an alternative to more traditional subword-based models. Tokenizer-free models are especially appealing to practitioners as they can eschew the two-step processing pipeline of subword segmentation and reduce the corresponding difficulties in cross-lingual transfer (Hu et al., 2020; Maronikolakis et al., 2021; Rust et al., 2021; Wang et al., 2021) or domain adaptation (Sato et al., 2020; Liu et al., 2021) due to inconsistent subword units.

However, upon several attempts to apply tokenizer-free methods, our analysis reveals several practical difficulties in applying these methods. This paper is a chronicle of some of the concerns we uncovered; we highlight some challenges with applying these models and propose best practices for future results reporting in this area.

Specifically, we perform experiments finetuning pretrained multilingual models, evaluating them with respect to finetuning data efficiency, inference time, and memory consumption. Based on these multiple dimensions, we come to the somewhat surprising conclusion that subword-based models, in particular mBERT (Devlin et al., 2019), might still be the most practical choice in most settings, as they perform best while maintaining a relatively low inference cost.

2 Tokenizer-free Multilingual Models

While multilingual pretrained models (Devlin et al., 2019; Lample and Conneau, 2019; Liu et al., 2020; Xue et al., 2021) have led to impressive performance improvements for low-resource languages through cross-lingual transfer, the standard word representation method in these models relies on subword segmentation (Sennrich et al., 2016; Kudo, 2018). In multilingual settings, subword tokenization can be sub-optimal as supporting hundreds of languages with various scripts and vocabularies causes segmentation mismatch between languages and over-segmentation in the lower-resourced languages (Wang et al., 2020).

To alleviate this problem, recent works propose removing the subword segmentation step by using characters or bytes as lexical units (Clark et al., 2022; Xue et al., 2022). In particular, these "tokenizer-free" methods have been applied to both encoder-only and encoder-decoder models. Tab. 1 presents an overview of the different tokenizer-free multilingual models with comparable subword models. Next, we briefly describe the two tokenizer-free models we consider in this work.

CANINE (Clark et al., 2022) is a character-level

[1]We will release code to train and evaluate models upon de-anonymization.

Findings of the Association for Computational Linguistics: EACL 2023, pages 1725–1735

Model	Params	Vocab (%)	Non-vocab	Architecture	Enc.	Dec.	Tokenization	↓sample?	Corpus	Langs
mBERT	178M	92M (52%)	86M	Enc-only	12	-	Subword	✗	Wikipedia	104
CANINE	132M	25M (19%)	107M	Enc-only	12	-	Character	✓	Wikipedia	104
mT5 (Small)	300M	256M (85%)	44M	Enc-dec	8	8	Subword	✗	mC4	101
ByT5 (Small)	300M	1.1M (0.3%)	298.5M	Enc-dec	12	4	UTF-8 bytes	✗	mC4	101

Table 1: Configuration of the pretrained models used. From left to right: number of parameters, number and ratio of vocabulary-related parameters, number of non-vocabulary parameters, architecture, encoder / decoder depth, tokenization scheme, whether downsampling was used, pretrained corpus, number of pretrained languages.

encoder suggested as an alternative to mBERT (Devlin et al., 2019). CANINE operates on raw characters and is pretrained using the masked language modeling objective. To compensate for the computational efficiency loss due to increased sequence length, CANINE uses convolutions to downsample the sequence before passing the representations to the transformer layers. The two weight variants of CANINE (CANINE-S, CANINE-C) have the same architecture but slightly different pretraining objectives using either subwords or characters at the last layer. As both variants performed similarly in our experiments and Clark et al. (2022), we only include CANINE-S for the main discussion, leaving CANINE-C results in § B.3.

ByT5 (Xue et al., 2022) is an encoder-decoder transformer model similar to the mT5 (Xue et al., 2021) model. ByT5 operates on the raw UTF-8 bytes of the input without any downsampling, leading to a longer sequence length while having a much smaller vocabulary size than mT5. Both ByT5 and mT5 are pretrained on the mC4 corpus[2] using the span reconstruction objective proposed by Raffel et al. (2020).

To keep the parameter count fixed between mT5 and ByT5, ByT5 allocates the parameters saved from the embedding layer to additional encoder layers. Although adding more depth to the encoder is a reasonable design choice, our results in § 4 show that ByT5 suffers from a much higher inference cost due to the deeper encoder, especially when input/output sequence lengths are longer.

3 Experimental settings

We conduct a multi-dimensional evaluation focusing on two aspects: finetuning data efficiency (§ 4.1) and inference cost (§ 4.2) to provide a better understanding of the practical applicability of tokenizer-free models. We finetune and evaluate two subword-based models (mBERT, mT5) and two tokenizer-free models (CANINE, ByT5), as mBERT-CANINE and mT5-ByT5 are directly comparable counterparts in terms of their pretraining corpus as shown in Tab. 1. For the T5 models, we consider only the small models of both mT5 and ByT5 as the focus of our work is in the practical implication of using multilingual pretrained models at relatively resource-constrained settings.

Specifically, we finetune the models on three multilingual natural language understanding tasks adopted from the XTREME benchmark (Hu et al., 2020). The three tasks we choose cover various input, output formats – sequence-level classification (XNLI), token-level classification (NER), and extractive question answering (TyDi QA-GoldP).

3.1 Tasks

XNLI The Cross-lingual Natural Language Inference (Conneau et al., 2018) is a sequence classification task in which the model predicts whether the hypothesis sentence is an entailment, contradiction, or neutral given the premise sentence. The task is provided in 15 languages.

NER Named Entity Recognition (NER) is a structured prediction task, where the model predicts a tag (location, person, organization) in IOB2 format for each token in the input sentence. We use the WikiAnn dataset (Pan et al., 2017) and select 20 out of 282 languages for multilingual training based on linguistic diversity and the language availability in the other two tasks we consider.

TyDi QA-GoldP The Typologically Diverse Question Answering (Clark et al., 2020) dataset is an extractive QA benchmark in 11 languages. While the original dataset includes two "primary" tasks (SelectP, MinSpan), the secondary GoldP task is the most widely adopted as it is compatible with other SQuAD-style QA tasks (Rajpurkar et al., 2016; Artetxe et al., 2020). For this reason, we mainly compare models on TyDi QA-GoldP

[2] https://www.tensorflow.org/datasets/catalog/c4#c4multilingual

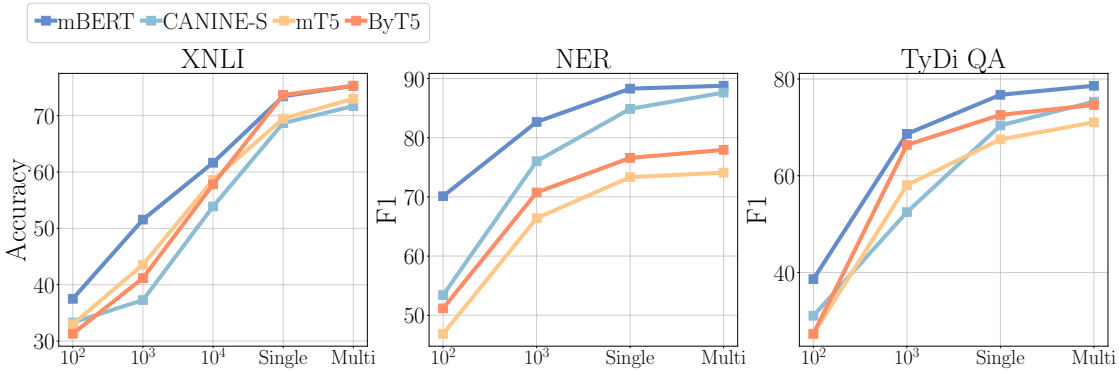

Figure 1: Average XNLI, NER, TyDi performance when each pretrained model is finetuned with varying numbers of in-language finetuning data (10^2, 10^3, 10^4), all in-language samples (Single), or the entire multilingual dataset (Multi). The exact numbers can be found in the Appendix (Tab. 2).

and discuss primary task results briefly through our replication experiment of Clark et al. (2022).

3.2 Details of Hardware and Measurements

We use a single Tesla V100 (32GB) GPU for all experiments regarding inference cost measurements. To obtain the peak GPU memory and inference latency, we randomly select 100 samples from the English test set for each task and measure the average cost of predicting one example at a time.

4 A Multi-dimensional Evaluation

4.1 Finetuning data efficiency

Most work presenting multilingual pretrained models evaluates downstream task performance under multilingual finetuning or zero-shot scenarios. In practice, however, downstream task datasets are often available in the language of interest. Thus, in addition to multilingual training, we compare models tuned on different data sizes *within* a single language to evaluate their finetuning data efficiency.

Specifically, we finetune the four pretrained models with varying numbers of task examples – 10^2, 10^3, 10^4 (when available), all target language samples (Single), and multilingual training (Multi) to incorporate situations where the task dataset is available in multiple languages. We experiment with four downstream task languages – English, Arabic, Russian, and Swahili – chosen based on both linguistic diversity and various pretraining resource conditions.[3] While the controlled experiments are done on a subset of languages, we report the task performance in all languages for zero-shot evaluation, single language training, and multilingual training in § B.3 for comprehensiveness.[4]

In Fig. 1, we report the models' task performance averaged over languages under different finetuning settings. Notably, we find that mBERT achieves the highest score for most settings. The only exception is on XNLI Single and Multi, where ByT5 slightly outperforms mBERT. As the dataset size decreases, it becomes more evident that mBERT is the most sample efficient, especially in the most data-scarce scenarios where only 100 finetuning examples are available. The fact that mBERT outperforms mT5 and ByT5 on smaller datasets is quite surprising, as one might expect T5 models to generalize better in low-resource settings given their much larger pretraining corpus.

Interestingly, we find that CANINE performs poorly compared to mBERT in all three tasks, and the performance gap increases as fewer finetuning data are available. To explain this phenomenon, we hypothesize that character-level models have the additional burden of learning to compose characters into semantically meaningful units and thus require more data to learn task-specific higher-level semantics. These results align with the NER results on the CoNLL and MasakhaNER dataset in Clark et al. (2022), where mBERT outperformed CANINE in all languages except Amharic, a language not covered by mBERT's vocabulary.

However, mBERT's stronger performance in TyDi QA-GoldP was unexpected as CANINE performed better at the TyDi QA primary tasks in Clark et al. (2022). Through replication experiments to reconcile the contradictory findings, we found that mBERT outperforms CANINE also in the primary tasks when finetuned for more epochs with our codebase, suggesting that the previous

[3]The pretraining corpus sizes are noted in § B.4 (Tab. 8).

[4]Hyperparameters for all experiments are in Appendix A.

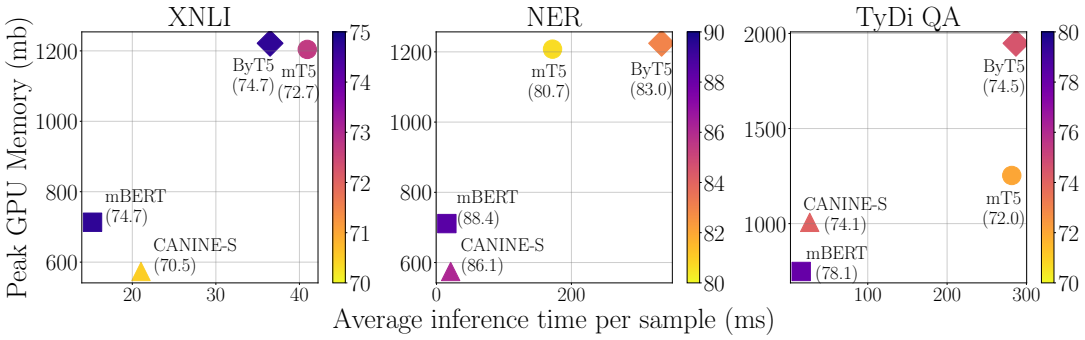

Figure 2: The inference cost of the four models (■: mBERT, ▲: CANINE, ●: mT5, ◆: ByT5) in each task. The x-axis denotes the average inference time while the y-axis shows the peak GPU memory consumption. Thus, models located near the bottom left corner are more cost efficient. The colors represent the model's best task performance (XNLI: Accuracy, NER: F1, TyDi QA: F1). The numbers used to generate the plot can be found in § B.2 (Tab. 3).

mBERT baseline was potentially undertrained.[5]

For mT5 and ByT5, we find that the two models perform comparably in smaller datasets, while on larger sets, ByT5 consistently outperforms mT5 on all tasks. We note that the mT5-Small model could have been penalized in terms of capacity as 85% of the parameters are allocated to embeddings as shown in Tab. 1, leaving only 44M parameters for the non-vocabulary layers. This is even less than that of mBERT (86M), and drastically smaller compared to ByT5-Small, which assigns 298.5M parameters to the non-vocabulary layers. Also, given that the tasks concerned are not generation-heavy, the extra depth on the encoder (12 for ByT5 vs. 8 for mT5) might have favored ByT5 over mT5.

4.2 Inference cost

Another key concern in utilizing pretrained models for downstream applications is the inference cost, such as memory consumption and latency. In Fig. 2, we plot each model's inference latency and peak memory consumption, color-coding their task performance to provide a comprehensive view of the trade-offs of deploying each model in practice.

In general, the encoder-only models, mBERT and CANINE, require much less memory and inference latency than mT5 and ByT5. Considering performance alongside inference cost, we find that mBERT is still the most practical choice among the four models, achieving the best performance while maintaining a relatively low inference cost.

While producing longer sequences than mBERT, CANINE does not necessarily incur higher memory or latency costs, as it has fewer parameters than mBERT. This helps CANINE, especially in sentence-level tasks (XNLI, NER) where inputs are relatively shorter. However, for tasks with much longer inputs (TyDi QA), the computational overhead from the sequence length dominates the parameter reduction, leading to higher memory usage and slower inference for CANINE.

For mT5 and ByT5, inference costs vary according to the task's input and output length. For tasks with shorter inputs and outputs like XNLI, ByT5 yields better performance than mT5 while retaining similar costs. However, for token-level prediction tasks like NER, ByT5 needs to generate tags autoregressively at the byte level, which drastically slows down the inference time. However, the additional cost is negligible in terms of memory consumption as the inputs are still relatively short. For TyDi QA, we observe an opposite pattern. As the input is a long passage, the extended input sequence significantly increases the memory consumption of ByT5, requiring more effort in tuning the batch size to fit into the GPU memory.

5 Related work

Large-scale NLP models have achieved remarkable performance in various natural language tasks, with the recent ChatGPT demonstrating near human-level language understanding capabilities. While achieving impressive results in standard benchmark settings, the applicability of these models have remained limited mainly due to practical considerations including their high energy consumption and environmental impact (Strubell et al., 2019). Both the NLP and computer vision communities have proposed evaluating models based on practical metrics, such as training/inference efficiency (Canziani et al., 2016; Dehghani et al., 2021; Zhou et al., 2021), energy usage (Henderson et al., 2020), robustness (Ribeiro et al., 2020; Kiela et al.,

[5] We include the finetuning code in our released codebase.

2021; Koh et al., 2021), and expected performance (Dodge et al., 2019). Similarly, a recent study by Liang et al. (2022) suggests a comprehensive evaluation suite for generative NLP models, including measures of robustness, fairness, and efficiency. Our multi-dimensional evaluation is an attempt to expand these evaluation protocols to *multilingual* settings and examine the trade-offs of various tokenization schemes.

6 Conclusion

In this paper, we present a multi-dimensional evaluation of tokenizer-free multilingual models focusing on their efficiency against finetuning dataset size and inference cost. Based on our experiments, we find that mBERT might still be the most cost-effective choice for many tasks, and show that the efficiency trade-offs of model design choices (tokenization, decoder availability) depend heavily on the task's length statistics. Despite our findings, tokenizer-free models still have a significant advantage in reducing engineering effort and potentially increasing robustness to noisy data. We believe more work should be done in developing *efficient* tokenizer-free models, and encourage the community to consider these criteria of practical applicability when developing and evaluating tokenizer-free pretrained models.

7 Limitations

This paper mainly covers three NLP tasks, focusing on smaller-sized multilingual pretrained models. In future work, it would be interesting to run the multi-dimensional evaluation we suggest on a broader set of tasks and models. Although our results show that subword models are a more practical choice in some tasks, we note that other tasks or datasets may exist where tokenizer-free methods achieve better relative performance. For instance, tokenizer-free models have been reported to excel in word-level tasks, and noisy environments (Xue et al., 2022), and the conclusions we reached may be different in such settings. Moreover, we did not explore more complicated generation tasks like translation or summarization, where the difficulty in decoding and longer decode horizons could paint a different picture in a multi-dimensional evaluation.

Ethics Statement

We hope our results encourage the community to consider the practical concerns of running large language models (LLMs) and designing tokenizer-free pretrained models. As the state-of-the-art LLMs are becoming more computationally extensive, it has become increasingly difficult for researchers and practitioners with less resources to utilize these models for downstream applications. We hope our multi-dimensional analysis can help researchers and practitioners with less computational resources decide which model to use in practice.

Acknowledgements

We acknowledge Kakao Enterprise for providing the compute resources for this work. We would like to thank Sanket Vaibhav Mehta, Daniel Fried, Saujas Vaduguru, and the anonymous reviewers for their valuable comments and feedback. Additionally, we would like to thank Jon Clark for answering questions related to the CANINE model. This work was supported in part by grant #2040926 from the National Science Foundation as well as the CMU-Portugal MAIA project.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *ACL*.

Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. 2016. An analysis of deep neural network models for practical applications.

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.

Jonathan H. Clark, Jennimaria Palomaki, Vitaly Nikolaev, Eunsol Choi, Dan Garrette, Michael Collins, and Tom Kwiatkowski. 2020. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Trans. Assoc. Comput. Linguistics*, 8:454–470.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. 2021. The efficiency misnomer. *CoRR*, abs/2110.12894.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. 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)*, pages 2185–2194, Hong Kong, China. Association for Computational Linguistics.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *J. Mach. Learn. Res.*, 21(248):1–43.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR.

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.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In *NeurIPS*.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D Manning, Christopher Ré, Diana Acosta-Navas, Drew A Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2022. Holistic evaluation of language models.

Xin Liu, Baosong Yang, Dayiheng Liu, Haibo Zhang, Weihua Luo, Min Zhang, Haiying Zhang, and Jinsong Su. 2021. Bridging subword gaps in pretrain-finetune paradigm for natural language generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6001–6011, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Antonis Maronikolakis, Philipp Dufter, and Hinrich Schütze. 2021. Wine is not v i n. on the compatibility of tokenizations across languages. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2382–2399, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1946–1958. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Shoetsu Sato, Jin Sakuma, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. 2020. Vocabulary adaptation for domain adaptation in neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4269–4279, Online. Association for Computational Linguistics.

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

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Xinyi Wang, Sebastian Ruder, and Graham Neubig. 2021. Multi-view subword regularization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, Online. Association for Computational Linguistics.

Zihan Wang, Karthikeyan K, Stephen Mayhew, and Dan Roth. 2020. Extending multilingual BERT to low-resource languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2649–2656, Online. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *NAACL*.

Xiyou Zhou, Zhiyu Chen, Xiaoyong Jin, and William Yang Wang. 2021. HULK: An energy efficiency benchmark platform for responsible natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 329–336, Online. Association for Computational Linguistics.

A Tasks

For all tasks and models, we refer to the original papers' codebase for hyperparameters.[6][7][8]

XNLI For encoder-only models, the first token (`[CLS]`) is used to map the sentence representation to the label distribution. For encoder-decoder models, we generate the index of the label (e.g., `'0'`) directly.

NER For encoder-decoder models, we follow the input-output format (e.g., input: `'tag: rick and morty are cool .'`, output: `'PER: rick $$ PER: morty'`) specified in the mT5 model's original codebase.

B Tables

B.1 Finetuning data efficiency

Tab. 2

B.2 Inference cost

Tab. 3

B.3 Experimental results for all languages (Zero-shot, Single language (full), Multilingual)

XNLI: Tab. 4, NER: Tab. 5, TyDi QA-GoldP: Tab. 6, Tydi QA Primary: Tab. 7

B.4 Pretraining corpus size

Tab. 8

[6] https://github.com/google-research/language/tree/master/language/canine

[7] https://github.com/google-research/multilingual-t5

[8] https://github.com/google-research/byt5

Finetuning setting	XNLI (Accuracy) 10^2	10^3	10^4	Single	Multi	NER (F1) 10^2	10^3	Single	Multi	TyDi QA (F1) 10^2	10^3	Single	Multi
Arabic													
mBERT	**36.6**	**51.0**	**59.5**	70.6	73.2	**67.3**	**80.2**	**89.6**	**89.6**	**44.8**	**70.9**	**81.0**	**81.5**
CANINE-S	32.8	36.6	53.3	65.8	69.7	46.2	71.2	84.9	88.0	38.4	59.8	79.2	80.5
CANINE-C	34.1	45.3	50.5	66.2	68.5	51.7	71.3	85.1	87.8	34.8	57.8	77.8	80.7
mT5	33.1	44.2	55.0	65.5	70.3	57.3	75.5	86.5	86.8	33.7	62.6	73.1	75.3
ByT5	23.7	42.0	55.2	**72.9**	**73.3**	60.6	77.5	85.4	87.7	33.4	67.3	75.8	75.9
English													
mBERT	**38.8**	**58.6**	**71.2**	**82.0**	**83.5**	**65.1**	**78.1**	**84.2**	**85.4**	**32.4**	**67.6**	**73.6**	**76.0**
CANINE-S	33.6	37.5	59.5	77.7	79.1	49.7	70.3	80.4	84.1	29.2	49.4	64.0	71.6
CANINE-C	34.1	50.7	61.2	77.1	78.0	52.8	70.6	81.1	84.1	27.5	47.8	57.3	71.6
mT5	33.3	50.9	66.4	79.0	79.9	40.1	63.1	71.9	72.5	25.0	52.8	59.4	64.4
ByT5	35.2	39.6	66.2	80.9	81.0	44.1	65.0	73.8	73.5	16.5	63.1	64.6	69.4
Russian													
mBERT	**35.7**	**45.5**	52.9	66.3	68.1	**81.3**	**89.9**	**90.0**	**90.9**	**42.9**	**74.3**	**79.8**	**82.4**
CANINE-S	33.1	35.9	48.6	61.5	65.0	63.2	86.9	87.7	89.6	29.1	54.0	71.3	77.4
CANINE-C	33.1	42.9	45.4	60.8	64.4	70.0	86.5	86.5	90.0	32.3	58.9	71.4	79.7
mT5	33.0	44.9	**58.0**	63.2	68.1	54.3	70.6	71.0	72.3	29.0	65.8	71.5	76.6
ByT5	34.3	41.2	56.4	**67.5**	**71.3**	68.6	83.5	84.5	84.3	32.5	73.5	78.8	80.3
Swahili													
mBERT	**38.9**	**51.1**	**63.0**	**74.8**	**76.4**	**66.7**	**82.4**	**89.4**	**89.2**	**34.3**	**61.8**	**72.5**	**74.4**
CANINE-S	33.7	39.2	54.2	69.7	73.0	54.5	75.6	86.5	88.8	27.4	46.6	67.2	71.8
CANINE-C	35.0	46.5	54.1	68.6	71.7	55.4	76.0	87.3	88.9	20.0	46.9	66.5	72.7
mT5	32.6	34.2	55.1	70.3	73.7	35.8	56.4	64.0	64.8	21.6	51.0	66.1	67.9
ByT5	32.0	42.0	53.4	73.4	75.6	31.4	57.0	62.6	66.3	26.9	61.6	71.1	73.0

Table 2: Task performance with varying finetuning data conditions ($10^2, 10^3, 10^4$ (for XNLI), full target language dataset, multilingual dataset)

	XNLI			NER			TyDi QA		
	Latency	Memory	Accuracy	Latency	Memory	F1	Latency	Memory	F1
mBERT	**15.24**	713.33	**74.7**	**15.30**	710.97	**88.4**	**16.10**	**748.34**	**78.13**
CANINE-S	21.04	**573.48**	70.5	20.96	**574.57**	86.1	26.89	1006.74	74.13
mT5	40.94	1204.19	72.7	171.99	1207.76	80.7	281.52	1253.13	72.05
ByT5	36.49	1221.54	**74.7**	333.72	1224.40	83.0	286.76	1948.30	74.48

Table 3: Inference latency (ms), peak GPU memory (mb), best average performance of each model in the three tasks

Model	en	ar	bg	de	el	es	fr	hi	ru	sw	th	tr	ur	vi	zh	avg
Zero-shot (en)																
mBERT	**82.0**	64.1	67.5	70.4	65.5	73.7	72.8	59.3	67.4	50.2	53.2	60.2	**57.5**	68.7	**68.1**	65.4
CANINE-S	77.7	50.1	60.1	62.4	53.7	67.6	66.0	43.7	60.7	40.4	39.6	47.9	41.1	53.1	43.2	53.8
CANINE-C	77.1	53.1	61.4	63.5	58.3	68.5	66.4	47.7	63.3	41.0	39.2	48.8	44.4	53.4	39.1	55.0
mT5-Small	79.0	61.3	66.0	64.4	67.4	65.9	62.4	**59.7**	66.6	52.2	**64.1**	57.9	56.4	57.3	63.9	63.0
ByT5-Small	80.9	**65.9**	**70.2**	**71.2**	**67.7**	**76.5**	**75.0**	58.6	**67.9**	**62.4**	58.4	**63.6**	55.6	**69.5**	64.9	**67.2**
Single-language																
mBERT	**82.0**	70.6	**76.2**	**76.6**	**75.1**	**77.7**	**77.4**	67.0	**74.8**	66.3	65.7	72.5	62.9	**75.9**	**76.4**	**73.1**
CANINE-S	77.7	65.8	70.6	72.4	68.6	73.8	73.4	61.2	69.7	61.5	59.9	66.6	58.0	67.4	57.2	66.9
CANINE-C	77.1	66.2	71.1	72.0	69.8	72.8	72.6	62.3	68.6	60.8	57.1	65.7	58.2	67.3	60.0	66.8
mT5-Small	79.0	65.4	69.9	72.0	73.6	73.1	74.8	65.2	70.3	63.2	69.7	67.6	58.9	69.2	71.0	69.5
ByT5-Small	80.9	**72.9**	75.4	75.8	**75.1**	**77.7**	76.4	**68.3**	73.4	**67.5**	**70.0**	**72.6**	**63.0**	72.7	72.5	73.0
Multilingual																
mBERT	**83.5**	73.2	77.7	**77.5**	75.7	**79.8**	**78.6**	**70.1**	**76.4**	68.1	67.2	**73.8**	64.4	**76.5**	**77.9**	**74.7**
CANINE-S	79.1	69.7	75.0	74.9	72.5	76.3	75.3	65.2	73.0	65.0	62.3	68.9	64.1	71.3	65.6	70.5
CANINE-C	78.0	68.5	73.7	74.1	72.9	75.7	74.9	63.8	71.7	64.4	57.7	67.9	62.6	69.7	58.7	69.0
mT5-Small	79.9	70.3	74.7	74.9	74.4	76.5	75.5	67.7	73.7	68.1	71.2	71.9	65.4	72.4	73.2	72.7
ByT5-Small	81.0	**73.3**	**77.8**	76.5	**76.5**	78.5	77.2	70.0	75.6	**71.3**	**71.4**	73.6	**68.3**	75.7	74.1	**74.7**

Table 4: XNLI Performance (Accuracy)

Model	en	ar	bn	de	el	es	fi	fr	hi	id	ja	ko	ru	sw	ta	te	th	tr	ur	zh	avg
Zero-shot (en)																					
mBERT	**84.2**	41.7	**68.2**	**78.2**	**71.4**	71.8	**77.3**	**78.0**	**64.5**	**51.6**	29.2	**59.7**	**65.6**	**71.4**	**51.0**	**50.4**	0.4	**73.9**	33.3	**43.1**	**58.2**
CANINE-S	80.8	29.6	49.6	70.7	63.5	66.4	66.7	74.1	41.1	47.3	0.5	29.3	57.7	59.8	28.4	19.7	0.1	55.8	22.0	5.4	43.4
CANINE-C	81.1	38.3	56.9	70.9	66.4	64.8	68.0	73.5	43.4	46.6	1.8	28.7	61.7	58.9	36.9	21.6	0.2	58.9	29.8	8.1	45.8
mT5-Small	71.9	32.9	56.6	67.1	42.3	70.0	65.1	75.3	56.2	45.3	25.5	23.9	36.9	49.0	38.0	35.9	3.6	58.7	**58.7**	31.3	47.2
ByT5-Small	73.8	**45.9**	61.5	70.7	67.7	**79.4**	67.1	77.4	57.1	46.2	**31.3**	26.2	46.7	60.2	31.9	27.9	**9.6**	23.3	1.3	32.8	46.9
Single-language																					
mBERT	**84.2**	**89.6**	**96.1**	**90.3**	**91.4**	**92.5**	**92.2**	**91.2**	**91.2**	**93.6**	**74.4**	**88.8**	**89.4**	**90.0**	**86.5**	**80.4**	**76.2**	**93.2**	**95.7**	**83.1**	**88.5**
CANINE-S	80.8	84.9	92.9	88.0	88.6	89.7	89.1	88.9	84.9	90.9	63.3	81.6	86.5	87.7	81.0	49.9	70.5	90.9	91.0	73.2	82.7
CANINE-C	81.1	85.1	93.5	87.5	89.1	89.8	88.4	88.4	84.3	90.6	60.2	79.5	87.3	86.5	79.6	43.0	74.0	90.6	92.4	68.9	82.0
mT5-Small	71.9	86.5	86.6	83.7	83.8	88.0	87.8	86.7	85.5	85.3	65.9	80.2	64.0	71.0	82.6	74.5	64.6	86.3	93.0	75.1	80.1
ByT5-Small	73.8	85.3	88.3	82.4	87.6	86.6	86.4	84.7	83.0	84.5	69.9	83.2	62.6	84.5	80.3	69.1	74.5	83.4	90.5	73.2	80.7
Multilingual																					
mBERT	**85.4**	**89.6**	**95.9**	**89.8**	**91.3**	**92.9**	**92.0**	**91.2**	**89.3**	**93.4**	**74.9**	**88.1**	**89.2**	**90.9**	**86.0**	**80.6**	76.5	**93.1**	**95.5**	**82.3**	**88.4**
CANINE-S	84.1	88.0	94.7	89.3	90.7	92.1	91.1	90.9	85.8	92.8	69.3	83.8	88.8	89.6	81.7	71.3	76.2	92.4	94.0	75.7	86.1
CANINE-C	84.1	87.8	95.6	89.2	91.1	92.5	90.7	90.9	88.2	92.6	67.9	81.5	88.9	90.0	81.6	69.5	**77.7**	92.0	93.7	72.1	85.9
mT5-Small	72.5	86.8	84.5	84.8	83.4	88.7	88.3	87.7	83.6	87.2	70.1	83.1	64.8	72.3	82.3	69.8	67.8	86.9	92.4	76.5	80.7
ByT5-Small	73.5	87.7	88.4	86.1	88.7	90.3	89.9	89.3	84.7	87.3	70.3	83.8	66.3	84.3	81.8	78.0	72.6	88.6	92.6	76.5	83.0

Table 5: NER Performance (F1)

Model	en	ar	bn	fi	id	ko	ru	sw	te	avg
Zero-shot (en)										
mBERT	**73.64**	**60.11**	**45.1**	**57.63**	**63.78**	**52.16**	**57.52**	**56.51**	**42.15**	**56.51**
CANINE-S	64.78	44.85	20.13	39.73	43.78	13.67	44.49	30.64	31.59	37.07
CANINE-C	63.96	42.19	22.05	43.13	36.87	17.44	42.02	33.3	30.51	36.83
mT5-Small	59.39	43.25	22.51	44.27	48.7	22.05	44.85	33.08	28.77	38.54
ByT5-Small	64.58	56.4	15.86	51.91	55.85	22.21	54.11	35.44	31.43	43.09
Single-language										
mBERT	**73.64**	**79.86**	**70.78**	**76.08**	79.93	**62.76**	**72.48**	**79.81**	81.21	**75.17**
CANINE-S	64.78	79.2	55.81	70.13	70.0	49.53	67.15	71.26	81.75	67.73
CANINE-C	63.96	77.79	50.92	67.28	66.26	49.84	66.49	71.39	82.78	66.3
mT5-Small	59.39	73.07	67.92	65.33	73.65	54.93	66.13	71.49	80.93	68.09
ByT5-Small	64.58	75.82	69.91	71.98	**80.55**	58.65	71.09	78.81	**85.39**	72.97
Multilingual										
mBERT	**76.02**	**81.49**	72.86	**80.41**	**84.87**	**67.09**	**74.45**	**82.42**	83.52	**78.13**
CANINE-S	71.55	80.53	67.24	75.42	78.44	61.25	71.75	77.43	83.53	74.13
CANINE-C	71.56	80.74	62.6	74.21	76.28	65.79	72.66	79.71	84.43	74.22
mT5-Small	64.39	75.34	**76.89**	70.01	76.73	59.24	67.86	76.62	81.35	72.05
ByT5-Small	69.42	75.86	70.9	74.52	79.78	60.62	73.01	80.32	**85.93**	74.48

Table 6: TyDi QA-GoldP Performance (F1)

Model	en	ar	bn	fi	id	ja	sw	ko	ru	te	th	avg
MINSPAN												
mBERT	**65.1**	**83.1**	**66.7**	**69.0**	**65.8**	**53.0**	**71.7**	**62.8**	**66.4**	**87.1**	**64.5**	**69.0**
CANINE-S	61.4	83.2	64.7	66.6	63.9	49.5	67.8	56.7	63.0	82.5	61.0	65.9
CANINE-C	58.8	82.6	58.7	64.7	64.3	50.8	65.1	56.2	64.4	83.9	61.5	65.2
SELECTP												
mBERT	**51.1**	**73.6**	**56.6**	**59.0**	**56.8**	**43.6**	**64.7**	**48.2**	**50.8**	**83.1**	**53.4**	**59.0**
CANINE-S	49.2	71.5	56.4	58.3	54.6	41.5	60.1	40.5	49.3	77.2	50.7	56.0
CANINE-C	47.4	71.0	46.5	53.8	54.4	40.2	56.0	34.0	48.8	78.0	49.1	53.2

Table 7: TyDi QA Primary Task Performance (F1)

Language	Wikipedia (Number of docs)	mC4 (Number of examples)
English	2.5M	3B
Russian	319K	756M
Arabic	77K	53M
Swahili	7K	985K

Table 8: Pretraining corpus sizes for languages used in § 4.1 experiments. The number of Wikipedia documents per language can be found here: https://en.wikipedia.org/wiki/Wikipedia:Multilingual_statistics