MzansiText and MzansiLM: An Open Corpus and Decoder-Only Language Model for South African Languages

Anri Lombard, Simbarashe Mawere, Temi Aina, Ethan Wolff, Sbonelo Gumede, Elan Novick, Francois Meyer, and Jan Buys
University of Cape Town
Cape Town, South Africa
{LMBANR001, MWRSIM003, ANXTEM001, WLFETH001, GMDSBO006, NVCELA001}@myuct.ac.za
{francois.meyer, jan.buys}@uct.ac.za

Abstract

Decoder-only language models can be adapted to diverse tasks through instruction finetuning, but the extent to which this generalizes at small scale for low-resource languages remains unclear. We focus on the languages of South Africa, where we are not aware of a publicly available decoder-only model that explicitly targets all eleven official written languages, nine of which are low-resource. We introduce MzansiText, a curated multilingual pretraining corpus with a reproducible filtering pipeline, and MzansiLM, a 125M-parameter language model trained from scratch. We evaluate MzansiLM on natural language understanding and generation using three adaptation regimes: monolingual task-specific finetuning, multilingual task-specific finetuning, and general multi-task instruction finetuning. Monolingual task-specific finetuning achieves strong performance on data-to-text generation, reaching 20.65 BLEU on isiXhosa and competing with encoder-decoder baselines over ten times larger. Multilingual task-specific finetuning benefits closely related languages on topic classification, achieving 78.5% macro-F1 on isiXhosa news classification. While MzansiLM adapts effectively to supervised NLU and NLG tasks, few-shot reasoning remains challenging at this model size, with performance near chance even for much larger decoder-only models. We release MzansiText and MzansiLM to provide a reproducible decoder-only baseline and clear guidance on adaptation strategies for South African languages at small scale.

Keywords: corpus creation, language modelling, language representation models, low-resource languages, multilinguality, natural language generation, finetuning, named entity recognition, text classification, South African languages

1. Introduction

The application of large language models (LLMs) to natural language processing tasks has progressed through three phases: pretrained models finetuned for specific tasks (BERT, T5) (Devlin et al., 2019; Raffel et al., 2020), in-context learning at inference time (GPT-3) (Brown et al., 2020), and instruction finetuning for broader generalization (Wei et al., 2022; Wang et al., 2023). Each phase has relied on increasingly larger models and more pretraining data, limiting direct applicability to low-resource settings.

For African languages, language model research has historically been led by encoder-only and encoder–decoder approaches (Ogueji et al., 2021; Alabi et al., 2022; Ogundepo et al., 2022; Adelani et al., 2022a), although decoder-only work has recently increased (Tonja et al., 2024a; Uemura et al., 2024; Buzaaba et al., 2025). Even so, decoder-only research for African languages remains comparatively early-stage, and many of the strongest established baselines are still encoder-only or encoder–decoder models. In this paper, we focus on the languages of South Africa, a country with particularly high linguistic diversity.

South Africa recognizes eleven official written languages: Sepedi (Northern Sotho), Sesotho, Setswana, siSwati, Tshivenda, Xitsonga, Afrikaans, English, isiNdebele, isiXhosa, and isiZulu. Afrikaans and English are Indo-European languages in the West Germanic branch, while the remaining nine are Bantu languages. Bantu languages are agglutinative and exhibit rich noun-class systems (Nurse and Philippson, 2004). Table 1 summarizes home-language usage and shows that only 8.7% of South Africans speak English at home (Statistics South Africa, 2024), which underscores the value of language technologies that work across the full set of official languages.

This paper introduces MzansiText and MzansiLM, a curated multilingual pretraining corpus and a 125M-parameter decoder-only model trained from scratch with pretraining coverage over all eleven official written South African languages. The dataset is available on HuggingFace at `anrilombard/mzansi-text` and the model at `anrilombard/mzansilm-125m`.

We systematically evaluate three adaptation regimes: task-specific monolingual finetuning, task-specific multilingual finetuning, and general multi-task instruction finetuning. MzansiLM achieves

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 1394–1408
11–16 May 2026.

Language	ISO	Speakers	%
isiZulu	zul	14,613,202	24.4
isiXhosa	xho	9,786,928	16.3
Afrikaans	afr	6,365,488	10.6
Sepedi (Northern Sotho)	nso	5,972,255	10.0
English	eng	5,228,301	8.7
Setswana	tsn	4,972,787	8.3
Sesotho	sot	4,678,964	7.8
Xitsonga	tso	2,784,279	4.7
siSwati	ssw	1,692,719	2.8
Tshivenda	ven	1,480,565	2.5
isiNdebele	nbl	1,044,377	1.7

Table 1: The number of L1 speakers for South Africa's eleven official written languages (Statistics South Africa, 2024).

competitive performance on some generation and sequence labelling tasks despite its modest parameter count, in some cases surpassing decoder-only models over ten times larger, though encoder-based architectures retain advantages on classification and other structured prediction tasks and few-shot reasoning remains challenging at this scale.

To summarise, this paper contributes:

- MzansiText, a curated pretraining corpus with a reproducible preparation pipeline and pretraining coverage over all eleven official written South African languages.
- MzansiLM, a 125M-parameter decoder-only model trained from scratch on MzansiText.
- A systematic comparison of different instruction finetuning approaches across available tasks for South African languages, revealing where decoder-only models excel and where they still fall short.

We also publicly release our code[1] for data preparation, pretraining, and evaluation.

2. Related Work

Current research in African NLP increasingly explores decoder-only architectures, though strong open-weight baselines are still predominantly encoder-only and encoder–decoder models. AfriBERTa (Ogueji et al., 2021) provides encoder-only baselines trained from scratch for several African languages. AfroXLMR (Alabi et al., 2022) adapts XLM-R (Conneau et al., 2020) to African languages through continued pretraining. On the encoder–decoder side, AfriTeVa (Ogundepo et al., 2022; Oladipo et al., 2023) builds sequence-to-sequence models by pretraining on the WURA corpus, which improves quality for African languages. Afri-mT5 (Adelani et al., 2022a) adapts mT5 (Xue et al., 2021) for African languages through continued pretraining. Aya-101 (Üstün et al., 2024) is an instruction-tuned encoder–decoder model covering 101 languages, providing a broad multilingual reference point.

A small but growing body of work investigates decoder-only models for African languages. InkubaLM (Tonja et al., 2024a) is a 0.4B parameter LM trained from scratch on a corpus that includes isiZulu, isiXhosa, Yoruba, Hausa, and Swahili, achieving competitive results on AfriXNLI and AfriMMLU, outperforming SmolLM-1.7B (Allal et al., 2024) and matching or exceeding LLaMA 3-8B (Dubey et al., 2024) on these tasks despite having far fewer parameters. AfroLlama-V1 Jacaranda Health et al. (2024) applies continual pretraining and instruction tuning to Llama-3-8B for Swahili, isiXhosa, isiZulu, Yoruba, Hausa, and English. Afri-Instruct (Uemura et al., 2024) continues pretraining LLaMA-2-7B on African languages, subsequently instruction tuning on diverse African tasks. Lugha-Llama (Buzaaba et al., 2025) adapts Llama-3.1-8B (Buzaaba et al., 2025) through continued pretraining for African languages.

Our work addresses three gaps. First, prior decoder-only models cover only subsets of South African languages (mainly Afrikaans, isiXhosa, isiZulu), never all eleven. Second, adaptation strategies are seldom compared: some focus on continued pretraining (Alabi et al., 2022; Ogundepo et al., 2022; Buzaaba et al., 2025), others on instruction tuning (Uemura et al., 2024), but systematic comparisons across task-specific monolingual, multilingual, and multi-task instruction finetuning remain absent. Third, evaluation centers on broad benchmarks that often exclude languages at the lower end of the resource-availability spectrum (IrokoBench (Adelani et al., 2025), AfriQA (Ogundepo et al., 2023)) rather than comprehensive South African language–specific datasets covering both understanding and generation.

3. Pretraining Dataset (MzansiText)

Training a decoder-only language model from scratch first requires a suitable pretraining corpus. We therefore construct MzansiText, a curated multilingual corpus for the eleven official written South African languages, by aggregating public sources and applying a reproducible preparation pipeline based on established best practices for large-scale web and multilingual corpora. This section introduces the data sources, preparation pipeline, and resulting token distribution.

Sources We draw on public language resources with substantive coverage or relevance to South African languages.

[1] github.com/Anri-Lombard/sallm

- mC4 (Raffel et al., 2020) consists of Common Crawl web text across 101 languages with filtering and deduplication.
- CulturaX (Nguyen et al., 2024) is a large multilingual dataset that applies a multi-stage language identification and cleaning process.
- WURA (Oladipo et al., 2023) cleans mC4 for African languages and combines it with targeted crawls of local news sites.
- Glot500-c (Imani et al., 2023) covers 511 languages, aggregating multiple sources and applying cleaning and deduplication.
- NCHLT Text (Eiselen and Puttkammer, 2014) provides curated monolingual corpora for South African languages developed through national initiatives. The material is smaller, but cleaner, than other web crawled corpora.
- CC100 (Wenzek et al., 2020) comprises monolingual data extracted from Common Crawl. It provides broad coverage but inconsistent quality, so further filtering is applied for our targets.
- ParaCrawl (ParaCrawl Project, 2020) is sentence-aligned parallel text mined from the web. It is useful as an indicator of in-language content and terminology, although alignment noise and domain skew are known challenges and are mitigated with strict filtering and deduplication. We use the monolingual African languages data drived from ParaCrawl for WMT2022.
- Inkuba-Mono (Lelapa AI - Fundamental Research Team, 2024) provides monolingual corpora used for the Inkuba family of LMs (Tonja et al., 2024b). It includes isiZulu and isiXhosa among five African languages, and is based on public web text and news sources for specific languages.

Language balance We split the pretraining data into 80%/10%/10% train, validation, and test splits. However, to prevent high-resource languages from dominating early stopping and model selection, the validation and test sets are capped at two million tokens per language. Unless otherwise stated, all token counts are computed with the 65,536-token tokenizer described in Section 4. Table 2 reports per-language token counts and within-split shares. Note that we use English data only from WURA, Glot500, and NCHLT Text.

Due to the resource imbalance between languages, the resulting distribution is heavily skewed toward Afrikaans and English (approximately 84% combined). We experimented with a more balanced pretraining mixture by capping English and Afrikaans to 25% each in the training data. However, this balanced version led to measurable degradation on downstream tasks (see Appendix A). We therefore retained the natural distribution to maximise usable high-quality data volume in this low-resource setting, while still ensuring pretraining coverage for every official language.

Preparation pipeline We implement a deterministic multi-stage pipeline with `Datatrove` to standardise quality across sources and languages (Hugging Face DataTrove Team, 2024). The design follows best practices reported for large web corpora such as C4/T5 (Raffel et al., 2020), Gopher (Rae et al., 2021), CulturaX (Nguyen et al., 2024), and FineWeb (Penedo et al., 2024):

1. Apply language identification at the document and line level, retaining segments whose dominant language is one of the eleven targets.
2. Normalise and remove boilerplate, including HTML stripping, Unicode normalisation, control characters, and excess whitespace.
3. Enforce structural constraints on length, script, and flag unexpected character distributions.
4. Remove exact and near-duplicate content using hashing and MinHash-style signatures, both within and across sources.
5. Apply safety screening to filter personally identifiable information and other sensitive content using conservative heuristics.
6. Apply heuristic quality filters to remove templated, boilerplate, and low-information text.
7. Pack short high-quality fragments into training records of roughly one thousand tokens while preserving sentence boundaries.

Table 3 reports token counts before and after deduplication and filtering. Across languages, deduplication removes 4.0 percent of the original word count, while subsequent filtering removes 22.9 percent of the deduplicated text, with the largest absolute reductions observed for Afrikaans, isiZulu, and isiXhosa.

4. Model Pretraining

MzansiLM is a compact decoder-only model trained on MzansiText. We use the architecture configuration from MobileLLM at 125M parameters (Liu et al., 2024), which employs hyperparameters optimized for small-scale models through systematic ablation studies, and adopt a LLaMA-style decoder stack with causal self-attention (Touvron et al., 2023). The maximum context length is 2,048. We train

Language	Train		Validation		Test	
	Tokens	%	Tokens	%	Tokens	%
Afrikaans	2,475,913,822	64.96	1,865,255	14.42	1,875,605	14.24
English	740,994,679	19.44	1,813,651	14.02	1,821,803	13.83
isiNdebele	818,549	0.02	106,224	0.82	143,458	1.09
Sepedi (Northern Sotho)	6,697,358	0.18	685,425	5.30	778,656	5.91
Sesotho	97,558,939	2.56	2,315,298	17.90	2,316,170	17.59
siSwati	1,932,989	0.05	196,247	1.52	225,810	1.71
Setswana	10,082,930	0.26	1,216,539	9.41	1,413,473	10.73
Xitsonga	3,013,408	0.08	510,463	3.95	319,496	2.43
Tshivenda	1,852,481	0.05	191,495	1.48	243,315	1.85
isiXhosa	152,212,403	3.99	2,016,503	15.59	2,012,000	15.28
isiZulu	320,224,015	8.40	2,017,406	15.60	2,021,343	15.35
Total	3,811,301,573	100.00	12,934,506	100.00	13,171,129	100.00

Table 2: Token counts by language and split after language identification and filtering using our 65,536-token tokenizer. Percentages show each language's share within a split. Validation and test apply a two-million-token per-language cap.

Source	Before processing	After deduplication	After filtering	Percent retained
WURA	997,742,420	988,157,747	879,523,389	88.2
mC4	1,008,467,039	979,623,283	824,839,355	81.8
CulturaX	702,050,710	695,083,559	676,035,198	96.3
Glot500	191,154,885	167,600,993	79,244,672	47.3
Inkuba-Mono	234,941,750	196,457,594	63,114,818	26.9
CC100	23,822,691	20,291,392	16,922,824	71.1
ParaCrawl	287,212,175	262,079,888	10,139,616	3.5
NCHLT Text	13,473,354	11,372,194	9,840,573	73.0

Table 3: Per-source word counts at three stages of the pipeline, aggregated across languages.

a BPE tokenizer (Sennrich et al., 2015) with a vocabulary size of 65,536 on our pretraining split (all dataset token counts reported in this paper are computed with this tokenizer).

Optimisation and scheduling follow the MobileLLM recipe at the 125M scale. We train for five epochs over the training split and monitor cross-entropy loss and mean token accuracy on the training and validation sets. To avoid higher-resource languages from dominating the held-out loss, we average held-out losses across the languages. All runs use four NVIDIA A100 accelerators with 80 GB of memory each. End-to-end pretraining completes in approximately 27 hours.

5. Instruction Finetuning

We adapt the base MzansiLM to several tasks. In our experiments we test three adaptation regimes that differ in their level of generalization: *monolingual task-specific finetuning* represents the narrowest scope, *multilingual task-specific finetuning* pools related languages under a shared task structure, and *multi-task instruction finetuning* combines all tasks and languages in a single training mixture. It remains unclear from the current literature which regime is most effective at this model scale and how this differs across languages and tasks. Prior work shows that the effectiveness of monolingual versus multilingual finetuning varies by task and language pair (Eisenschlos et al., 2019; Tang et al., 2021), while multi-task instruction tuning introduces additional tradeoffs between task coverage and task-specific performance (Uemura et al., 2024).

5.1. Datasets

We identified Natural Language Understanding and Generation datasets covering multiple South African languages which are suited for instruction finetuning. Table 4 reports train, validation, and test counts for the supervised datasets in our main instruction-tuning mixture. We use existing splits when provided; otherwise, we create a stratified validation split from the training data for early stopping.

Document classification MasakhaNEWS (Adelani et al., 2023) and SIB-200 (Adelani et al., 2024) provide document-level topic classification. MasakhaNEWS contains news articles across seven news categories while SIB-200 contains sentences labelled into seven categories. INJONGO

Task	Language	Train		Validation		Test	
		Examples	Tokens	Examples	Tokens	Examples	Tokens
AfriHG	Xho	10,440	5,892,814	1,305	750,506	1,305	734,845
	Zul	14,209	7,495,625	1,777	952,525	1,776	944,750
T2X	Xho	3,859	346,518	460	44,208	378	44,296
INJONGO Intent	Eng	1,779	398,170	622	139,312	622	139,312
	Sot	2,240	501,824	320	71,720	640	143,370
	Xho	2,240	509,035	320	72,795	640	145,383
	Zul	2,240	508,096	320	72,624	640	145,139
MasakhaNER 2.0	Tsn	3,489	609,722	499	87,620	996	173,044
	Xho	5,718	920,599	817	142,658	1,633	274,254
	Zul	5,848	922,761	836	134,701	1,670	266,979
MasakhaNEWS	Eng	3,309	2,594,818	472	369,539	948	752,019
	Xho	1,032	607,254	147	84,010	297	173,229
SIB-200	Afr	701	78,995	99	10,851	204	22,691
	Eng	701	77,191	99	10,635	204	22,267
	Nso	701	91,974	99	12,468	204	26,812
	Sot	701	87,709	99	12,064	204	25,394
	Xho	701	81,956	99	11,281	204	23,645
	Zul	701	81,336	99	11,198	204	23,429
TOTAL	–	**60,609**	**21,806,397**	**8,939**	**3,216,760**	**14,573**	**4,974,308**

Table 4: Supervised data used for instruction tuning, reported per task and language. Counts include both prompt and completion tokens.

Intent (Yu et al., 2025) targets intent classification with 40 intent classes across five domains.

Sequence labelling For sequence labelling, we use MasakhaPOS (Dione et al., 2023) for part-of-speech tagging and MasakhaNER 2.0 (Adelani et al., 2022b) for named entity recognition.

Generation AfriHG (Ogunremi et al., 2024) evaluates summarization via headline generation for isiXhosa and isiZulu, while T2X (Meyer and Buys, 2024) assesses data-to-text generation (mapping triples to descriptive sentences) in isiXhosa. The availability of suitable generation datasets for most of the target languages remain very limited.

Reading comprehension and few-shot understanding Belebele (Bandarkar et al., 2024) evaluates multiple-choice reading comprehension. IrokoBench (Adelani et al., 2025) is a benchmark suite for evaluating few-shot reasoning on African languages, comprising three tasks: AfriXNLI for natural language inference, AfriMMLU for knowledge-based multiple-choice questions, and AfriMGSM for mathematical word problems. These tasks are challenging for even large-scale LMs, many of which exhibit chance performance (Adelani et al., 2025).

5.2. Finetuning Setup

We format all supervised tasks as next-token prediction, preserving the decoder-only architecture without introducing encoder components or task-specific heads. Inputs use a maximum length of 2,048 tokens for classification and labelling and 1,024 tokens for generation. We apply early stopping on each task's validation split and select checkpoints by the task's primary validation metric, using token accuracy for sequence labelling and task-standard metrics elsewhere. When multiple instruction templates are available, training cycles through them to cover the formats encountered in zero-, one-, and three-shot prompting.

For most tasks, we use the prompt templates from LM Evaluation Harness (EleutherAI, 2024) to ensure consistency between training and evaluation formats. For T2X and AfriHG, we use standard templates that prompt the model with an input and request the corresponding output.

We employ LoRA for parameter-efficient finetuning with rank 128, scaling $\alpha = 256$, and dropout 0.1, applied to attention and feed-forward projections. Most tasks have limited training data and models overfit without parameter-efficient adaptation such as LoRA. We conducted Bayesian hyperparameter tuning and use settings adapted from pretraining unless a task requires a shorter training horizon. We set the number of finetuning epochs for each task: MasakhaNEWS, INJONGO Intent, and MasakhaPOS train for 20 epochs, SIB-200 for 15 epochs, MasakhaNER 2.0 for 50 epochs, AfriHG for 3 epochs, and T2X for 4 epochs.

We compare the following adaptation approaches of our base MzansiLM:

- **Base 0-shot:** no finetuning and no in-context examples at evaluation.

Model / Variant	Size	BLEU	chrF	ROUGE-L
T2X (isiXhosa)				
MzansiLM				
Base 0-shot	0.125B	0.00	0.03	3.29
Base 1-shot	0.125B	0.00	0.03	4.08
Base 3-shot	0.125B	0.00	0.00	0.74
mono-t2x-ft	0.125B	**20.65**	**31.56**	**41.19**
general-ft	0.125B	0.00	0.00	2.05
Encoder–Decoder				
mT5-base	0.58B	16.8	28.7	38.7
Aya	13B	8.9	22.1	33.9
AfriHG (isiXhosa)				
MzansiLM				
Base 0-shot	0.125B	0.00	0.03	3.29
Base 1-shot	0.125B	0.00	0.03	4.08
Base 3-shot	0.125B	0.00	0.00	0.74
mono-afrihg-ft	0.125B	0.63	4.47	14.28
multi-afrihg-ft	0.125B	**0.95**	5.67	**16.98**
general-ft	0.125B	0.36	4.10	12.42
Encoder–Decoder				
AfriTeVa2 base	0.313B	–	14.9	–
mT5-base	0.58B	–	12.7	–
Aya	13B	–	**15.2**	–
AfriHG (isiZulu)				
MzansiLM				
Base 0-shot	0.125B	0.00	0.07	4.24
Base 1-shot	0.125B	0.00	0.07	4.61
Base 3-shot	0.125B	0.00	0.29	0.86
mono-afrihg-ft	0.125B	**1.33**	9.67	**19.05**
multi-afrihg-ft	0.125B	0.79	7.05	17.48
general-ft	0.125B	0.00	3.34	10.08
Encoder–Decoder				
AfriTeVa2 base	0.313B	–	**17.4**	–
mT5-base	0.58B	–	15.5	–
Aya	13B	–	16.2	–

Table 5: Combined results for isiXhosa and isiZulu across T2X (data-to-text) and AfriHG (headline generation). Boldface indicates best performance; underline indicates second best.

- **Base few-shot:** no finetuning but 1 or 3 in-context examples at evaluation.
- **Monolingual task-specific finetuning:** finetune a separate model for each task–language pair using only examples from that language and task.
- **Multilingual task-specific finetuning:** finetune one model per task by pooling all languages with examples for that task, enabling cross-lingual transfer in a fixed label space.
- **Multi-task instruction finetuning:** finetune a single model on all tasks and languages jointly using a weighted sampling mixture.

In the multi-task regime we use fixed, per-dataset sampling weights to balance supervision across the task types. Across all regimes, we reuse the tokenizer described in Section 4. Test-time decoding follows task-appropriate settings. Results in

Model / Variant	Size	Eng	Xho
MzansiLM			
Base 0-shot	0.125B	38.3	39.1
mono-masakhanews-ft	0.125B	63.5	73.2
multi-masakhanews-ft	0.125B	60.8	78.5
general-ft	0.125B	49.2	50.9
Decoder–Only			
InkubaLM-0.4B	0.4B	20.3	7.4
AfroLlama-V1	8B	67.1	50.2
Llama-3.1-70B-Instruct	70B	83.3	68.4
Encoder–Decoder			
Aya-101	13B	87.1	94.6
Encoder–Only			
AfriBERTa	0.126B	88.9	87.0
AfroXLMR-base	0.270B	92.2	94.7
AfroXLMR-large	0.550B	**93.1**	**97.3**

Table 6: Macro-F1 scores for MasakhaNEWS topic classification. Boldface marks the best performance, while underline marks the second best.

Model / Variant	Size	Xho	Zul	Tsn
MzansiLM				
Base 0-shot	0.125B	0.0	0.0	0.0
mono-masakhaner-ft	0.125B	48.4	26.1	38.9
multi-masakhaner-ft	0.125B	24.8	22.6	23.7
general-ft	0.125B	13.6	20.7	12.3
Decoder–Only				
InkubaLM-0.4B	0.4B	0.1	0.0	0.0
AfroLlama-V1	8B	5.6	3.8	3.4
Llama-3.1-70B-Instruct	70B	8.2	10.1	20.7
Encoder–Decoder				
Aya-101	13B	0.0	0.0	0.0
Encoder–Only				
AfriBERTa	0.126B	85.0	81.7	83.2
AfroXLMR-base	0.270B	88.6	88.4	87.7
AfroXLMR-large	0.550B	**89.9**	**90.6**	**89.4**

Table 7: Macro-F1 scores for MasakhaNER 2.0 named entity recognition. Boldface marks the best performance, while underline marks the second best.

Tables 5 through 11 come directly from the three regimes defined here.

6. Results

We evaluate MzansiLM and its finetuned variants across all tasks listed in Section 5.1. The results are presented in Tables 5–11. This section addresses two research questions. First, how do the three finetuning strategies—monolingual task-specific, multilingual task-specific, and general multi-task instruction—compare to each other? Second, how competitive is MzansiLM at the 125M scale against larger baseline models?

Model / Variant	Size	Xho	Zul	Tsn
MzansiLM				
Base 0-shot	0.125B	0.0	0.0	0.0
mono-masakhapos-ft	0.125B	30.5	37.8	5.6
multi-masakhapos-ft	0.125B	11.6	9.5	0.5
general-ft	0.125B	0.0	0.0	0.0
Decoder–Only				
InkubaLM-0.4B	0.4B	0.0	0.0	0.0
AfroLlama-V1	8B	0.0	0.0	0.0
Llama-3.1-8B-Instruct	8B	58.1	64.2	49.5
Llama-3.1-70B-Instruct	70B	68.6	71.6	52.2
Encoder–Decoder				
Aya-101	13B	0.0	0.0	0.0
Encoder–Only				
AfriBERTa	0.126B	86.1	86.9	82.5
AfroXLMR-base	0.270B	88.5	89.4	82.7
AfroXLMR-large	0.550B	**88.7**	**90.1**	**83.0**

Table 8: Token accuracy for MasakhaPOS part-of-speech tagging. Boldface marks the best performance, while underline marks the second best.

6.1. Comparing Finetuning Strategies

All finetuning regimes raise performance well above base model prompting for document classification. On MasakhaNEWS (Table 6), multilingual task-specific finetuning achieves the highest macro-F1 in isiXhosa, benefiting from shared signal across closely related Bantu languages, while monolingual task-specific finetuning is competitive in English. On SIB-200 (Table 9), both monolingual and multilingual task-specific finetuning improve accuracy relative to base prompting, but the leading variant varies by language and no single regime dominates. INJONGO Intent performance (Table 9) remains near chance (2.5%) across all finetuning regimes at this model size. For sequence labelling (Tables 7 and 8), base model prompting does not produce usable output. Monolingual task-specific finetuning is the most effective adaptation regime, while multilingual task-specific finetuning and general multi-task finetuning are weaker at this model scale.

Monolingual task-specific finetuning performs decisively stronger than base prompting and general multi-task finetuning on isiXhosa data-to-text generation (Table 5), with the gains indicating that targeted supervision is essential at this scale. Headline generation in isiXhosa and isiZulu remains challenging, with base model prompting near zero. Among the finetuned variants, multilingual task-specific finetuning is strongest for isiXhosa while monolingual task-specific finetuning leads in isiZulu (Table 5).

Normalized accuracy on Belebele reading comprehension (Table 10) clusters near chance for South African languages under base prompting. General multi-task finetuning improves accuracy on English and Afrikaans but not the target languages. On the IrokoBench evaluation suite for few-shot reasoning (Table 11) general multi-task finetuning does not improve performance over base prompting at this scale.

6.2. Comparison to Baseline Models

MzansiLM demonstrates competitive performance against larger models on several specific tasks. Monolingual task-specific finetuning on T2X data-to-text generation reaches 20.65 BLEU on isiXhosa, surpassing mT5-base and competing with encoder–decoder baselines over ten times larger (Table 5). On named entity recognition, monolingual task-specific finetuning substantially outperforms all other decoder-only baselines, including InkubaLM at 0.4B parameters, AfroLlama-V1 at 8B parameters, and even Llama-3.1-70B-Instruct, demonstrating that task-specific supervision at small scale can exceed few-shot prompting of much larger models (Table 7). Multilingual task-specific finetuning achieves 78.5% macro-F1 in isiXhosa on MasakhaNEWS topic classification, outperforming both InkubaLM and AfroLlama-V1 (Table 6). However, MzansiLM remains less competitive on part-of-speech tagging and named entity recognition, with meaningful gains only from monolingual task-specific finetuning and substantially stronger performance from larger prompted or encoder-based baselines (Tables 8 and 7).

On tasks where MzansiLM achieves near-chance performance, it matches or exceeds other decoder-only models at comparable or larger scales, indicating that these tasks are fundamentally challenging for decoder-only architectures below 70B parameters. On INJONGO Intent, MzansiLM's performance is comparable to InkubaLM at 0.4B and AfroLlama-V1 at 8B, with only Llama-3.1-70B-Instruct showing substantial improvement (Table 9). Similarly, MzansiLM's normalized accuracy on Belebele reading comprehension matches InkubaLM and AfroLlama-V1 across South African languages, with all decoder-only models below 70B parameters remaining near chance (Table 10). On AfriXNLI, AfriMMLU, and AfriMGSM, the pattern repeats: MzansiLM performance aligns with decoder-only models up to 8B parameters, and substantial gains require scaling to 70B (Table 11).

Encoder-only models retain a clear advantage on high-accuracy classification and sequence labelling tasks. AfroXLMR-large consistently achieves macro-F1 scores above 89 on both MasakhaNEWS and MasakhaNER 2.0, substantially ahead of MzansiLM's finetuned variants (Tables 6 and 7). On SIB-200 and INJONGO Intent, encoder-only baselines reach accuracy in the 80-90 range while all decoder-only models remain far below, with INJONGO Intent particularly challenging for decoder-only architectures at any scale below 70B (Table 9). Encoder–decoder models dominate read-

Task	Model / Variant	Size	Eng	Xho	Zul	Sot	Nso	Afr
SIB-200	**MzansiLM**							
	Base 0-shot	0.125B	32.3	28.9	31.2	18.0	17.3	43.4
	mono-sib200-ft	0.125B	28.0	39.1	22.0	36.8	36.6	54.4
	multi-sib200-ft	0.125B	33.3	40.4	47.2	34.7	30.5	29.6
	general-ft	0.125B	28.0	39.1	47.2	36.8	33.9	54.4
	Decoder–Only							
	InkubaLM-0.4B	0.4B	9.0	8.4	8.2	5.3	6.4	5.3
	AfroLlama-V1	8B	6.4	6.5	6.4	39.7	38.7	6.4
	Llama-3.1-70B-Instruct	70B	88.3	65.0	57.3	54.4	55.8	85.6
	Encoder–Decoder							
	Aya-101	13B	82.8	82.0	82.9	81.4	82.1	83.7
	Encoder–Only							
	AfriBERTa	0.126B	–	70.7	73.5	55.9	54.8	89.8
	AfroXLMR-base	0.270B	–	83.1	84.9	**83.7**	80.7	90.4
	AfroXLMR-large	0.550B	–	**84.0**	**85.8**	83.5	**83.3**	**91.1**
INJONGO Intent	**MzansiLM**							
	Base 0-shot	0.125B	3.1	0.8	0.9	1.4	–	–
	mono-injongo-intent-ft	0.125B	3.4	0.6	0.6	0.9	–	–
	multi-injongo-intent-ft	0.125B	2.9	1.0	0.7	0.7	–	–
	general-ft	0.125B	3.5	0.6	0.6	0.9	–	–
	Decoder–Only							
	InkubaLM-0.4B	0.4B	0.4	0.3	0.3	0.3	–	–
	AfroLlama-V1	8B	10.1	1.8	1.0	0.1	–	–
	Llama-3.1-70B-Instruct	70B	84.7	34.3	35.2	23.3	–	–
	Encoder–Decoder							
	Aya-101	13B	70.7	62.7	57.2	45.9	–	–
	Encoder–Only							
	AfriBERTa	0.126B	74.2	94.4	95.0	81.9	–	–
	AfroXLMR-base	0.270B	84.1	**97.3**	96.1	85.5	–	–
	AfroXLMR-large	0.550B	**84.5**	96.9	**97.7**	**86.8**	–	–

Table 9: Accuracy for SIB-200 (topic classification) and INJONGO Intent (intent classification). Boldface marks the best performance, while underline marks the second best.

Model / Variant	Size	Xho	Zul	Tsn	Ssw	Sot	Eng	Afr
MzansiLM								
Base 0-shot	0.125B	27.8	28.0	28.3	27.8	27.1	27.3	27.3
Base 1-shot	0.125B	29.4	28.2	30.8	29.7	27.4	27.4	27.6
Base 3-shot	0.125B	28.8	28.2	29.0	28.8	27.3	27.6	27.4
general-ft	0.125B	21.9	25.1	21.9	23.6	22.0	29.6	31.1
Decoder–Only								
InkubaLM-0.4B	0.4B	23.1	23.2	24.6	–	–	23.9	25.9
AfroLlama-V1	8B	24.8	28.2	26.8	22.9	22.6	25.3	24.7
Llama-3.1-8B-Instruct	8B	35.1	35.3	32.3	32.3	33.7	80.7	66.9
Meta-Llama-3-70B-Instruct	70B	41.3	42.9	41.4	42.9	51.3	**93.2**	**88.9**
Encoder–Decoder								
Aya-101	13B	**65.9**	**64.9**	**63.6**	**57.6**	**61.7**	86.1	81.7
Encoder–Only								
XLM-V large	–	54.4	54.2	–	47.1	32.7	77.8	72.3

Table 10: Normalised accuracy for Belebele reading comprehension. Boldface marks the best performance, while underline marks the second best.

ing comprehension and headline generation. Aya-101 at 13B parameters achieves normalized accuracy of 65.9 on Belebele isiXhosa, far exceeding all decoder-only models including those at 70B scale (Table 10). Similarly, AfriTeVa V2 and Aya-101 substantially outperform MzansiLM on AfriHG headline generation (Table 5).

Overall, these results show that task-specific supervision is most effective for generation and sequence labelling at 125M parameters, that multilingual task-specific finetuning can help closely related Bantu languages on topic classification, and that encoder-based models remain preferable for more structured prediction tasks. Few-shot

		AfriXNLI			AfriMMLU				AfriMGSM			
Model / Variant	**Size**	**Xho**	**Zul**	**Sot**	**Xho**	**Zul**	**Sot**	**Eng**	**Xho**	**Zul**	**Sot**	**Eng**
MzansiLM												
Base 0-shot	0.125B	32.3	32.0	32.3	22.7	24.7	23.7	25.4	1.6	2.4	1.2	0.8
general-ft	0.125B	33.4	33.6	31.8	23.0	24.2	25.1	23.4	2.4	0.8	1.6	1.6
Decoder–Only												
InkubaLM-0.4B	0.4B	34.3	34.2	33.5	26.2	23.4	26.6	24.3	9.2	9.2	2.4	12.0
AfroLlama-V1	8B	41.8	38.3	34.3	26.6	27.1	26.0	34.0	0.0	0.4	0.8	0.4
Llama-3.1-8B-Instruct	8B	34.0	34.7	34.8	27.2	31.8	31.8	62.8	4.4	5.2	6.4	56.8
Llama-3.1-70B-Instruct	70B	39.3	37.0	39.0	**34.2**	**40.4**	**39.0**	**76.4**	**18.8**	**27.2**	**32.8**	**86.8**
Encoder–Decoder												
Aya-101	13B	**55.2**	**55.3**	**54.7**	32.2	31.2	33.2	42.8	4.4	4.8	10.8	11.6

Table 11: Accuracy for IrokoBench instruction following tasks: AfriXNLI, AfriMMLU, and AfriMGSM. Boldface marks the best performance, while underline marks the second best.

multiple-choice reasoning, reading comprehension, and intent classification are difficult for decoder-only models below 70B parameters, indicating that these tasks require further scaling. For some languages, this is not possible, in which case encoder-only and encoder–decoder architectures remain the more viable and data-efficient choice.

7. Conclusion

This paper introduced MzansiText and MzansiLM, establishing an open, reproducible decoder-only LM for all eleven written South African languages. Evaluations across three adaptation regimes show that task-specific supervision is most effective at this scale, while few-shot reasoning and reading comprehension remain challenging below 70B parameters. Our findings clarify the role of small-scale decoder-only LMs for low-resource languages. Rather than replacing encoder-only and encoder–decoder models on tasks where those architectures remain stronger, such as classification and structured prediction, MzansiLM serves as a reproducible decoder-only baseline that helps identify where task-specific finetuning is effective and where alternative architectures are still preferable. At the same time, our results show that smaller models trained from scratch with sound design choices and region-specific language coverage can still match or even outperform much larger models on structured generation tasks. MzansiText, MzansiLM, and our comprehensive set of results provide a benchmark for future research on language modelling for South African languages.

Limitations

We train a 125M-parameter model, which aligns with scaling law expectations given our pretraining data volume and computational budget. This scale enables reproducibility and rapid iteration but constrains our ability to evaluate emergent capabilities such as few-shot reasoning and long-range dependency modeling. The model's token budget limits prompt design for multi-example inputs, and performance on tasks requiring substantial reasoning capacity remains near chance. These constraints are consistent with the scaling behavior observed across decoder-only models at this parameter count and reflect our focus on establishing a reproducible baseline rather than achieving state-of-the-art performance on all tasks.

Data composition is a second limitation. The pretraining mixture is heavily skewed toward Afrikaans and English. As detailed in Section 3, we experimented with a more balanced corpus but observed measurable degradation on downstream tasks and therefore retained the natural distribution.

While MzansiText provides pretraining coverage for all eleven official languages, our downstream evaluation focuses on the eight languages for which we identified the most directly comparable public benchmark coverage for the tasks considered in this paper. Public evaluation resources do exist for isiNdebele, Tshivenda, and Xitsonga, but their coverage is more fragmented across tasks and does not align as cleanly with the benchmark suite used here.

The general instruction-tuning mixture introduces trade-offs: combining tasks and languages improves coverage but dilutes task-specific signal, particularly for low-resource generative tasks. Evaluation scope is a further constraint. We focus on non-translation tasks and adopt automated metrics, which provide comparability but do not fully capture fluency, error severity, or output quality.

Ethical Considerations

All datasets used in this study are publicly available and were collected from open multilingual resources. No private or user-generated data were included. The released model is small in scale, trained for research purposes, and unlikely to cause harm or misuse. We encourage responsible use and clear citation of the datasets and models.

Acknowledgements

This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Number 151601) and the Telkom–National Research Foundation (Telkom-NRF) Future Technologies Programme. Computations were performed using facilities provided by the University of Cape Town's ICTS High Performance Computing team: https://hpc.uct.ac.za

8. References

David Ifeoluwa Adelani, Jesujoba Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen Muhammad, Guyo Jarso, Oreen Yousuf, Andre Niyongabo Rubungo, Gilles Hacheme, Eric Peter Wairagala, Muhammad Umair Nasir, Benjamin Ajibade, Tunde Ajayi, Yvonne Gitau, Jade Abbott, Mohamed Ahmed, Millicent Ochieng, Anuoluwapo Aremu, Perez Ogayo, Jonathan Mukiibi, Fatoumata Ouoba Kabore, Godson Kalipe, Derguene Mbaye, Allahsera Auguste Tapo, Victoire Memdjokam Koagne, Edwin Munkoh-Buabeng, Valencia Wagner, Idris Abdulmumin, Ayodele Awokoya, Happy Buzaaba, Blessing Sibanda, and Andiswa Bukula. 2022a. A few thousand translations go a long way! Leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. 2024. Sib-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta. Association for Computational Linguistics.

David Ifeoluwa Adelani, Marek Masiak, Israel Abebe Azime, Jesujoba Alabi, Atnafu Lambebo Tonja, Christine Mwase, Odunayo Ogundepo, Bonaventure F. P. Dossou, Akintunde Oladipo, Doreen Nixdorf, Chris Chinenye Emezue, Sana Al-azzawi, Blessing Sibanda, Davis David, Lolwethu Ndolela, Jonathan Mukiibi, Tunde Ajayi, Tatiana Moteu, Brian Odhiambo, Abraham Owodunni, Nnaemeka Obiefuna, Muhidin Mohamed, Shamsuddeen Hassan Muhammad, Teshome Mulugeta Ababu, Saheed Abdullahi Salahudeen, Mesay Gemeda Yigezu, Tajuddeen Gwadabe, Idris Abdulmumin, Mahlet Taye, Oluwabusayo Awoyomi, Iyanuoluwa Shode, Tolulope Adelani, Habiba Abdulganiyu, Abdul-Hakeem Omotayo, Adetola Adeeko, Abeeb Afolabi, Anuoluwapo Aremu, Olanrewaju Samuel, Clemencia Siro, Wangari Kimotho, Onyekachi Ogbu, Chinedu Mbonu, Chiamaka Chukwuneke, Samuel Fanijo, Jessica Ojo, Oyinkansola Awosan, Tadesse Kebede, Toadoum Sari Sakayo, Pamela Nyatsine, Freedmore Sidume, Oreen Yousuf, Mardiyyah Oduwole, Kanda Tshinu, Ussen Kimanuka, Thina Diko, Siyanda Nxakama, Sinodos Nigusse, Abdulmejid Johar, Shafie Mohamed, Fuad Mire Hassan, Moges Ahmed Mehamed, Evrard Ngabire, Jules Jules, Ivan Ssenkungu, and Pontus Stenetorp. 2023. Masakhanews: News topic classification for african languages. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 144–159, Nusa Dua, Bali. Association for Computational Linguistics.

David Ifeoluwa Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba O. Alabi, Shamsuddeen H. Muhammad, Peter Nabende, Cheikh M. Bamba Dione, Andiswa Bukula, Rooweither Mabuya, Bonaventure F. P. Dossou, Blessing Sibanda, Happy Buzaaba, Jonathan Mukiibi, Godson Kalipe, Derguene Mbaye, Amelia Taylor, Fatoumata Kabore, Chris Chinenye Emezue, Anuoluwapo Aremu, Perez Ogayo, Catherine Gitau, Edwin Munkoh-Buabeng, Victoire Memdjokam Koagne, Allahsera Auguste Tapo, Tebogo Macucwa, Vukosi Marivate, Elvis Mboning, Tajuddeen Gwadabe, Tosin Adewumi, Orevaoghene Ahia, Joyce Nakatumba-Nabende, Neo L. Mokono, Ignatius Ezeani, Chiamaka Chukwuneke, Mofetoluwa Adeyemi, Gilles Q. Hacheme, Idris Abdulmumin, Odunayo Ogundepo, Oreen Yousuf, Tatiana Moteu Ngoli, and Dietrich Klakow. 2022b. Masakhaner 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4488–4508, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Ifeoluwa Adelani, Jessica Ojo, Israel Abebe

Azime, Jian Yun Zhuang, Jesujoba Oluwadara Alabi, Xuanli He, Millicent Ochieng, Sara Hooker, Andiswa Bukula, En-Shiun Annie Lee, Chiamaka Ijeoma Chukwuneke, Happy Buzaaba, Blessing Kudzaishe Sibanda, Godson Koffi Kalipe, Jonathan Mukiibi, Salomon Kabongo Kabenamualu, Foutse Yuehgoh, Mmasibidi Setaka, Lolwethu Ndolela, Nkiruka Odu, Rooweither Mabuya, Salomey Osei, Shamsuddeen Hassan Muhammad, Sokhar Samb, Tadesse Kebede Guge, Tombekai Vangoni Sherman, and Pontus Stenetorp. 2025. Irokobench: A new benchmark for african languages in the age of large language models. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2732–2757, Albuquerque, New Mexico. Association for Computational Linguistics.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to african languages via multilingual adaptive fine-tuning. In *Proceedings of the AfricaNLP Workshop at ICLR*.

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Leandro von Werra, and Thomas Wolf. 2024. Smollm - blazingly fast and remarkably powerful.

Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. The belebele benchmark: a parallel reading comprehension dataset in 122 language variants. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Happy Buzaaba, Alexander Wettig, David Ifeoluwa Adelani, and Christiane Fellbaum. 2025. Lugha-llama: Adapting large language models for african languages. *CoRR*, abs/2504.06536. DBLP entry for the arXiv version.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Cheikh M. Bamba Dione, David Ifeoluwa Adelani, Peter Nabende, Jesujoba Alabi, Thapelo Sindane, Happy Buzaaba, Shamsuddeen Hassan Muhammad, Chris Chinenye Emezue, Perez Ogayo, Anuoluwapo Aremu, Catherine Gitau, Derguene Mbaye, Jonathan Mukiibi, Blessing Sibanda, Bonaventure F. P. Dossou, Andiswa Bukula, Rooweither Mabuya, Allahsera Auguste Tapo, Edwin Munkoh-Buabeng, Victoire Memdjokam Koagne, Fatoumata Ouoba Kabore, Amelia Taylor, Godson Kalipe, Tebogo Macucwa, Vukosi Marivate, Tajuddeen Gwadabe, Mboning Tchiaze Elvis, Ikechukwu Onyenwe, Gratien Atindogbe, Tolulope Adelani, Idris Akinade, Olanrewaju Samuel, Marien Nahimana, Théogène Musabeyezu, Emile Niyomutabazi, Ester Chimhenga, Kudzai Gotosa, Patrick Mizha, Apelete Agbolo, Seydou Traore, Chinedu Uchechukwu, Aliyu Yusuf, Muhammad Abdullahi, and Dietrich Klakow. 2023. MasakhaPOS: Part-of-speech tagging for typologically diverse African languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10883–10900, Toronto, Canada. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song,

Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurull, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, et al. 2024. The llama 3 herd of models.

Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard. 2019. MultiFiT: Efficient multi-lingual language model fine-tuning. 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 5702–5707, Hong Kong, China. Association for Computational Linguistics.

EleutherAI. 2024. lm-evaluation-harness (v0.4.3).

Hugging Face DataTrove Team. 2024. Datatrove: Large-scale data processing for llm training. Open-source library for processing, filtering, and deduplicating web-scale text.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. 2024. Mobilellm: Optimizing sub-billion parameter language models for on-device use cases.

Francois Meyer and Jan Buys. 2024. Triples-to-isixhosa (t2x): Addressing the challenges of low-resource agglutinative data-to-text generation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16841–16854, Torino, Italia. ELRA and ICCL.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2024. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. In *Proceedings of the Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4226–4237, Torino, Italy. ELRA and ICCL.

Derek Nurse and Gérard Philippson, editors. 2004. *The Bantu Languages*. Routledge. Overview of Bantu typology, including noun-class systems.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Odunayo Ogundepo, Tajuddeen R. Gwadabe, Clara E. Rivera, Jonathan H. Clark, Sebastian Ruder, David Ifeoluwa Adelani, Bonaventure F. P. Dossou, Abdou Aziz Diop, Claytone Sikasote, Gilles Hacheme, Happy Buzaaba, Ignatius Ezeani, Rooweither Mabuya, Salomey Osei, Chris Emezue, Albert Njoroge Kahira, Shamsuddeen Hassan Muhammad, Akintunde Oladipo, Abraham Toluwase Owodunni, Atnafu Lambebo Tonja, Iyanuoluwa Shode, Akari Asai, Tunde Oluwaseyi Ajayi, Clemencia Siro, Steven Arthur, Mofetoluwa Adeyemi, Orevaoghene Ahia, Anuoluwapo Aremu, Oyinkansola Awosan, Chiamaka Chukwuneke, Bernard Opoku, Awokoya Ayodele, Verrah Otiende, Christine Mwase, Boyd Sinkala, Andre Niyongabo Rubungo, Daniel A. Ajisafe, Emeka Felix Onwuegbuzia, Habib Mbow, Emile Niyomutabazi, Eunice Mukonde, Falalu Ibrahim Lawan, Ibrahim Said Ahmad, Jesujoba O. Alabi, Martin Namukombo, Mbonu Chinedu, Mofya Phiri, Neo Putini, Ndumiso Mngoma, Priscilla A. Amouk, Ruqayya Nasir Iro, and Sonia Adhiambo. 2023. AfriQA: Cross-lingual open-retrieval question answering for African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14957–14972, Singapore. Association for Computational Linguistics.

Odunayo Ogundepo, Akintunde Oladipo, Mofetoluwa Adeyemi, Kelechi Ogueji, and Jimmy Lin. 2022. Afriteva: Extending "small data" pretraining approaches to sequence-to-sequence models. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language*

Processing, pages 126–135, Dublin, Ireland. Association for Computational Linguistics.

Toyib Ogunremi, Serah Akojenu, Anthony Soronnadi, Olubayo Adekanmbi, and David Ifeoluwa Adelani. 2024. Afrihg: News headline generation for african languages. OpenReview.

Akintunde Oladipo, Mofetoluwa Adeyemi, Orevaoghene Ahia, Abraham Toluwalase Owodunni, Odunayo Ogundepo, David Ifeoluwa Adelani, and Jimmy Lin. 2023. Better quality pre-training data and t5 models for african languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)*, pages 158–168, Singapore. Association for Computational Linguistics.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*.

Jack W. Rae et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Statistics South Africa. 2024. Census 2022 in brief. Technical report, Stats SA. Table 3.8 & Figure 3.9.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Atnafu Lambebo Tonja, Bonaventure F. P. Dossou, Jessica Ojo, Jenalea Rajab, Fadel Thior, Eric Peter Wairagala, Aremu Anuoluwapo, Pelonomi Moiloa, Jade Abbott, Vukosi Marivate, and Benjamin Rosman. 2024a. InkubaLM: A small language model for low-resource african languages. *arXiv preprint arXiv:2408.17024*.

Atnafu Lambebo Tonja, Bonaventure F. P. Dossou, Jessica Ojo, Jenalea Rajab, Fadel Thior, Eric Peter Wairagala, Anuoluwapo Aremu, Pelonomi Moiloa, Jade Abbott, Vukosi Marivate, and Benjamin Rosman. 2024b. Inkubalm: A small language model for low-resource african languages. *arXiv*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Kosei Uemura, Mahe Chen, Alex Pejovic, Chika Maduabuchi, Yifei Sun, and En-Shiun Annie Lee. 2024. AfriInstruct: Instruction tuning of african languages for diverse tasks. In *Findings of EMNLP*, pages 13571–13585.

Ahmet Üstün, Viraat Aryabumi, Zheng Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya Model: An instruction finetuned open-access multilingual language model. In *Proceedings of ACL*, pages 15894–15939.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of ACL*, pages 13484–13508.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*, Virtual Conference.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. 2020. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, pages 4003–4012, Marseille, France. European Language Resources Association.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively

multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Hao Yu, Jesujoba Oluwadara Alabi, Andiswa Bukula, Jian Yun Zhuang, En-Shiun Annie Lee, Tadesse Kebede Guge, Israel Abebe Azime, Happy Buzaaba, Blessing Kudzaishe Sibanda, Godson Koffi Kalipe, Jonathan Mukiibi, Salomon Kabongo Kabenamualu, Mmasibidi Setaka, Lolwethu Ndolela, Nkiruka Odu, Rooweither Mabuya, Shamsuddeen Hassan Muhammad, Salomey Osei, Sokhar Samb, Dietrich Klakow, and David Ifeoluwa Adelani. 2025. Injongo: A multicultural intent detection and slot-filling dataset for 16 african languages. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9429–9452, Vienna, Austria. Association for Computational Linguistics.

9. Language Resource References

Eiselen, Roald and Puttkammer, Martin. 2014. *SADiLaR NCHLT Text Corpora*. SADiLaR Language Resource Management Agency. SADiLaR. PID https://hdl.handle.net/20.500.12185/1.

Imani, Ayyoob and Lin, Peiqin and Kargaran, Amir Hossein and Severini, Silvia and Jalili Sabet, Masoud and Kassner, Nora and Ma, Chunlan and Schmid, Helmut and Martins, André and Yvon, François and Schütze, Hinrich. 2023. *Glot500-c*. CIS, LMU Munich. Hugging Face. PID https://huggingface.co/datasets/cis-lmu/Glot500.

Jacaranda Health and Stanslaus, Mwongela and Jay, Patel and Rajasekharan, Lusiji Sathy and Lyvia and Francesco, Piccino and Mfoniso, Ukwak and Ellen, Sebastian. 2024. *AfroLlama V1: An instruction-tuned Llama model for African languages*. Hugging Face. PID https://huggingface.co/Jacaranda/AfroLlama_V1.

Lelapa AI - Fundamental Research Team. 2024. *Inkuba-Mono: Monolingual Corpora for African Languages*. Lelapa AI. Hugging Face. PID https://huggingface.co/datasets/lelapa/Inkuba-Mono. Dataset accompanying InkubaLM.

Nguyen, Thuat and Nguyen, Chien Van and Lai, Viet Dac and Man, Hieu and Ngo, Nghia Trung and Dernoncourt, Franck and Rossi, Ryan A. and Nguyen, Thien Huu. 2024. *CulturaX*. UONLP. Hugging Face. PID https://huggingface.co/datasets/uonlp/CulturaX.

Oladipo, Akintunde and Adeyemi, Mofetoluwa and Ahia, Orevaoghene and Owodunni, Abraham Toluwalase and Ogundepo, Odunayo and Adelani, David Ifeoluwa and Lin, Jimmy. 2023. *WURA*. Castorini. Hugging Face. PID https://huggingface.co/datasets/castorini/wura.

ParaCrawl Project. 2020. *ParaCrawl Release v7.1*. ParaCrawl Project. ParaCrawl. PID https://paracrawl.eu/v7-1.

A. Balanced Pretraining Mixture

As noted in Section 3, we also experimented with a more balanced pretraining mixture that reduced the dominance of Afrikaans and English and redistributed more capacity to the nine Bantu languages. We evaluated this balanced-pretraining variant on a subset of downstream tasks to test whether a less skewed corpus would improve adaptation for lower-resource languages.

The results did not support that hypothesis. On MasakhaNEWS and MasakhaNER 2.0, the balanced-pretraining variant generally underperformed the main model trained on the natural data distribution. On SIB-200, INJONGO Intent, Belebele, AfriXNLI, AfriMMLU, and AfriMGSM, performance remained near chance and did not materially differ from the main model. We did not evaluate the balanced-pretraining variant on the generation tasks at that stage, so we do not draw conclusions for AfriHG or T2X.

These exploratory results are consistent with our decision to retain the natural distribution for the main experiments: in this low-resource setting, preserving access to the largest available volume of higher-quality text was more beneficial than enforcing a more balanced language mixture.

Task	Variant	Eng	Xho	Zul	Sot	Tsn
MasakhaNEWS (Macro-F1)	mono-ft	55.4	64.8	–	–	–
	multi-ft	53.1	68.7	–	–	–
MasakhaNER 2.0 (Macro-F1)	mono-ft	–	39.2	18.7	–	29.8
	multi-ft	–	17.4	15.9	–	16.8
SIB-200 (Accuracy)	Base 0-shot	31.1	27.6	29.8	17.5	16.9
	general-ft	27.4	36.8	44.9	34.1	29.8
INJONGO Intent (Accuracy)	Base 0-shot	3.0	0.7	0.8	1.2	–
	general-ft	3.2	0.6	0.6	0.8	–
Belebele (Norm. Acc.)	Base 0-shot	26.9	27.1	27.4	26.8	27.7
	general-ft	28.7	21.4	24.5	21.7	21.0
AfriXNLI (Accuracy)	Base 0-shot	–	31.9	31.5	31.7	–
	general-ft	–	32.6	32.8	31.1	–
AfriMMLU (Accuracy)	Base 0-shot	24.8	22.1	24.0	23.1	–
	general-ft	23.0	22.6	23.7	24.4	–
AfriMGSM (Accuracy)	Base 0-shot	0.8	1.2	2.0	1.2	–
	general-ft	1.2	2.0	0.8	1.2	–

Table 12: Exploratory downstream results for the balanced-pretraining variant of MzansiLM. The model was evaluated on a subset of downstream tasks only; generation tasks were not included for this variant.