Swahili - Sukuma Machine Translation

Zephania Reuben Isack Odero

Technical report · August 2026 · Research preview

Quality SKU: https://huggingface.co/nileagi/nileagi-suk-mt

Lite SKU: https://huggingface.co/nileagi/nileagi-suk-mt-lite

Abstract

This report describes NileAGI's two public Swahili-to-Sukuma translators, `nileagi/nileagi-suk-mt` (quality) and `nileagi/nileagi-suk-mt-lite` (lite). Both are encoder–decoder sequence-to-sequence systems trained on 31,102 literary sentence pairs, split by whole document groups into 29,962 / 993 / 147 for train, validation, and test. Decode always forces a Sukuma start token. The quality SKU is the checkpoint with the best held-out chrF2; the lite SKU is a smaller sibling trained on the same table. On the document-group test set, quality reaches chrF2 **42.6** and BLEU **20.0**; lite reaches chrF2 **34.8** and BLEU **12.7**. Scores are a research preview, not production translation. The systems are one-way. English is not a source language: English input must be pivoted through Swahili.

1 Introduction

Sukuma (Kisukuma, ISO 639-3 `suk`) is a Bantu language of northern Tanzania. Speakers routinely use Swahili as the regional written bridge, so Swahili–Sukuma bitext is the practical way to put Sukuma into a translation stack. Public parallel data for this pair is still scarce. Everyday conversation, news, chat, and social media are not represented in the NileAGI table, and this report does not claim coverage of those domains.

NileAGI's MT SKUs do one job: **Swahili text in, Sukuma text out**, in a Latin orthography with vowel macrons (`ā ē ī ō ū`). There is no public Sukuma-to-Swahili SKU and no English-to-Sukuma SKU. An English cascade is English → Swahili with any public system, then one of the two models below.

We publish two operating points on the same data and recipe, not two different tasks:

Table 1: Public NileAGI Sukuma MT SKUs.

SKU	Hub repo	Role
Quality	`nileagi/nileagi-suk-mt`	Best test chrF2; default choice
Lite	`nileagi/nileagi-suk-mt-lite`	Same table and recipe; smaller footprint

Both live in collection https://huggingface.co/collections/nileagi/nileagi-suk. This document is only about the two translators.

The rest of the report is organised as follows. Section 2 states the task, direction, and orthography. Section 3 describes the parallel table and the document-group splits. Section 4 gives the architecture

class, tokenizer work, decoding constraint, and optimisation. Section 5 reports validation and test metrics. Section 6 discusses qualitative behaviour. Section 7 covers inference, intended use, and licence.

2 Task and language notes

2.1 Direction

The only supported mapping is Swahili → Sukuma. The public tokenizers mark the source language as `swh_Latn` and the target beginning-of-sequence token as `suk_Latn`. Both must be set at inference. If either is omitted, the decoder does not produce Sukuma.

2.2 Script and spelling

Output is Latin script. NileAGI keeps vowel macrons. Two written Sukuma practices exist in the wild (with and without diacritics). Models trained on this table will look like domain shift on macron-free text, and the reverse is also true: users who strip macrons from the output are no longer matching the reference orthography used in evaluation.

2.3 Length

On the NileAGI table, Sukuma is longer than Swahili: 733,283 target words against 587,871 source words, or about 1.25× whitespace tokens per sentence (means 23.6 vs. 18.9). Character-level metrics are therefore a better primary score than word-level BLEU.

2.4 Register

The table is literary written language, produced as read-aloud text rather than dialogue. Greetings and short formulaic lines are in-domain only insofar as they appear in that register. Chat, legal drafting, clinical notes, and breaking news are out of scope.

2.5 English

English never appears as a source field. A user who starts from English should treat Swahili as an intermediate language. Errors in that first hop are copied into Sukuma; this report does not score the cascade.

3 Data

All MT supervision is a literary Swahili–Sukuma sentence table of 31,102 pairs, released under the Nwulite Obodo Open Data License (NOODL) 1.0. The table is text pairs only. It is not redistributed with the MT checkpoints.

Table 2: Parallel table used for both MT SKUs. Word counts are whitespace tokens.

Field	Value
Source language	Swahili (`sw`)
Target language	Sukuma (`suk`)
Sentence pairs	31,102
Document groups	66
Sections	1,189
Swahili words	587,871 (∼18.9 / sentence)
Sukuma words	733,283 (∼23.6 / sentence)
Unique whitespace types	Swahili 63,251 Sukuma 77,438
Register	literary written language
Orthography	Latin with macrons
Split rule	whole document group

3.1 Inventory

Columns of the training table are `src_text`, `tgt_text`, `src_lang` (always `sw`), `tgt_lang` (always `suk`), `split`, `doc_id` (anonymised document group), and `unit_id` (unit inside the document).

3.2 Why document-group splits

Random sentence splits leak. Neighbouring units from the same document share names, constructions, and discourse. We therefore hold out *entire document groups*. A sentence in test never has a neighbour from the same group in train. The cost is a small test set: enough to rank two SKUs and to quote a headline chrF2, not enough for a fine-grained error taxonomy by genre.

Table 3: Document-group splits. Share is pair count over 31,102.

Split	Pairs	Share	Role
train	29,962	96.3%	parameter updates
validation	993	3.2%	checkpoint selection (chrF2)
test	147	0.5%	headline numbers in this report

Test chrF2 is the number to quote. Validation is only for choosing a checkpoint and for a sanity check that training is not collapsing. If the held-out groups change, the scores will move. That is expected for a document-level split of this size.

4 Models and training

Both SKUs are encoder–decoder sequence-to-sequence translators. Training updates the full network. The two SKUs differ in capacity and in a few optimisation details so that each fits the same recipe; they are not two different tasks.

4.1 Decoding constraint

At every evaluation and in the public inference snippet, generation is started with the Sukuma language token `suk_Latn`. The source tokenizer language is `swh_Latn`. This pair is part of the

public API. Users who call `generate` without `forced_bos_token_id` will not get the behaviour reported in Section 5.

4.2 Target-side vocabulary

Sukuma macrons and stems are poorly covered by a generic multilingual tokenizer. On the **training** Sukuma side only, we fit a unigram SentencePiece model (vocab size 2,000, character coverage 1.0) and add every piece that is missing from the existing tokenizer. **1,426** pieces were added for both SKUs. The embedding matrix is resized to the new vocabulary. If a dedicated Sukuma language token is new, its embedding is copied from the Swahili language token so that forced BOS has a defined vector. This step is identical for quality and lite.

4.3 Optimisation

The objective is label-smoothed sequence-to-sequence cross-entropy, using Hugging Face `Seq2SeqTrainer` defaults unless listed below. Mixed precision is FP16. Gradient checkpointing is on. The learning rate is 1×10^{-4}. The effective batch is 16 sequences. At evaluation, generation is capped at 128 new tokens with forced Sukuma BOS. The trainer keeps the checkpoint with the highest **validation chrF2**.

Table 4: Shared training settings.

Setting	Value
Objective	label-smoothed seq2seq cross-entropy
Learning rate	1×10^{-4}
Effective batch	16 sequences
Precision	FP16
Gradient checkpointing	yes
Eval generation	max 128 new tokens, forced Sukuma BOS
Selection rule	highest validation chrF2

Table 5: SKU-specific training settings. Both SKUs see the same 29,962 training pairs.

	Quality	Lite
Epochs	3	2
Source/target truncate	160 tokens	192 tokens
Optimizer steps	5,619	3,746

The quality SKU uses a slightly shorter truncate so the larger network fits the same training recipe. Lite stops after two epochs because validation chrF2 has already been used to pick its shipped checkpoint. Neither SKU uses the test split for early stopping.

5 Evaluation

5.1 Protocol

All headline numbers come from Hugging Face `Seq2SeqTrainer.evaluate` with `predict_with_generate`, forced target BOS, and a maximum of 128 new tokens.

- **chrF2** (`evaluate` / `chrf`): character n-grams, $\beta = 2$, scale 0–100. This is the **primary** metric.
- **BLEU** (`evaluate` / `bleu`, tokenizer default): reported $\times 100$ to match common tables. Secondary.
- **Eval loss**: token-level loss on teacher-forced labels, not on sampled text.

BLEU is a poor fit for this pair. Sukuma is morphologically rich, about 1.25× longer than Swahili on this table, and decoding sometimes inserts spaces inside words. A system can look worse on BLEU than a human reading of the same string. We still report BLEU for comparability with other MT tables. We do not rank the two SKUs by BLEU.

5.2 Validation

Validation (993 pairs) is the selection set. Table 6 shows chrF2 and BLEU after each epoch. Bold marks the shipped checkpoint for that SKU; lite is shipped after epoch 2, so it has no epoch-3 row.

Table 6: Document-group validation (993 pairs). Bold: selected checkpoint per SKU. Lite ships at epoch 2 and was not trained further.

Epoch	Quality chrF2	Quality BLEU	Lite chrF2	Lite BLEU
1	39.1	16.2	34.0	10.6
2	44.6	21.4	**39.1**	**16.3**
3	**46.3**	**23.2**	n/a	n/a

Quality keeps improving through epoch 3 (chrF2 39.1 → 44.6 → 46.3). Lite is shipped at epoch 2 (chrF2 34.0 → 39.1). The quality–lite gap on validation is about 7 chrF2 at the selected checkpoints (46.3 vs. 39.1).

5.3 Test

Table 7 is the table to quote. The test set is 147 pairs from held-out document groups. Bold marks the primary metric (chrF2).

Table 7: Document-group test (147 pairs). Headline numbers for this release. Bold: primary metric.

Metric	Quality	Lite
chrF2	**42.6**	**34.8**
BLEU	20.0	12.7
eval loss	1.97	2.46

Quality is ahead on every test number: +7.8 chrF2, +7.3 BLEU, and lower loss. Lite is the SKU to load when the quality checkpoint is too heavy for the deployment, not when the best string is required.

The drop from validation to test (quality 46.3 → 42.6 chrF2; lite 39.1 → 34.8) is expected. Test is a handful of whole documents, not a large random sample. Do not over-interpret a one-point chrF2 change on this split.

A chrF2 in the mid-30s to low-40s means the output often carries the meaning but is not a clean match to the reference. That is a research preview, not a substitute for a professional translator.

6 Qualitative behaviour

Short Swahili greetings often land in plausible Sukuma. On the quality SKU, `Habari yako?` decodes as `Ūlī mhola?`. Formulaic thanks and welcome lines are in the same band: usable, not guaranteed.

Longer literary sentences are less stable. Names and rare stems are often approximated rather than copied. Because the table is literary, conversational particles and code-switching with English or Swahili slang are not modelled.

A recurring defect is **spaces inside Sukuma words**. Extra spaces inflate word-level mismatch and make BLEU look worse than character overlap (chrF2). Downstream display should not treat every space as a morphological boundary. When scoring new text, prefer chrF2.

7 Using the models

7.1 Inference

Set the source language and force the Sukuma BOS token. The snippet below is the quality SKU; replace the repo id with `nileagi/nileagi-suk-mt-lite` for lite.

Listing 1: Inference example (Python)

```python
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer

repo = "nileagi/nileagi-suk-mt"
tok = AutoTokenizer.from_pretrained(repo)
model = AutoModelForSeq2SeqLM.from_pretrained(repo)

tok.src_lang = "swh_Latn"
inputs = tok("Habari yako?", return_tensors="pt")
bos = tok.convert_tokens_to_ids("suk_Latn")
out = model.generate(
    **inputs,
    forced_bos_token_id=bos,
    max_new_tokens=128,
)
print(tok.decode(out[0], skip_special_tokens=True))
```

Generation in this report used `max_new_tokens=128`. Longer documents should be split into sentences or short paragraphs before decode. There is no public English tokenizer path: translate English to Swahili first.

7.2 Which SKU

Use quality unless the runtime cannot hold it. Lite is trained on the same pairs, with the same tokenizer extension and the same forced BOS, and it is behind on every reported metric. Do not mix the two in a single document if you need consistent spelling and spacing.

7.3 Intended use

- Literary Swahili → Sukuma.
- Last hop of English → Swahili → Sukuma.

7.4 Out of scope

- Sukuma → Swahili.
- Direct English → Sukuma.
- Chat, legal, medical, or news translation.
- Conversational or social-media Sukuma.
- Commercial products without a licence from NileAGI.

7.5 Limitations

1. **Preview scores.** Test chrF2 in the 30s–40s is usable for research, not a replacement for a human translator.
2. **Register.** Spoken Sukuma and other orthographies will look like domain shift.
3. **Test size.** 147 document-group pairs. Headline numbers are stable enough to choose quality over lite, not to claim a precise human-parity gap.
4. **Spacing.** Intra-word spaces hurt BLEU and some tokenizers.
5. **Pivot error.** Any English cascade copies Swahili MT errors into Sukuma.
6. **Rare stems.** Names and low-count types are often approximated.

7.6 Licence

Table 8: Licences. The data licence is not the licence of the weights.

Artifact	Licence
`nileagi-suk-mt` and `nileagi-suk-mt-lite` weights	CC BY-NC-SA 4.0
Parallel training table	NOODL 1.0

Non-commercial use of the weights with attribution and share-alike. Commercial use of the weights needs a written agreement with NileAGI.

8 Conclusion

NileAGI releases two one-way Swahili → Sukuma translators trained on a document-group split of 31,102 literary pairs. Quote **test chrF2**: 42.6 for quality, 34.8 for lite. Use quality when the string matters; use lite when the footprint matters. Forced BOS `suk_Latn` and source language

`swh_Latn` are required. The systems are a research preview on a literary macron orthography, not a general Sukuma product.

9 Citation

Listing 2: BibTeX entry

```bibtex
@misc{nileagi-suk-mt-techreport-2026,
  title        = {NileAGI Sukuma Machine Translation: Technical Report},
  author       = {NileAGI},
  year         = {2026},
  howpublished = {Hugging Face},
  url          = {https://huggingface.co/nileagi/nileagi-suk-mt}
}
```