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 translation systems,nileagi/nileagi-suk-mt (quality) and `nileagi/nileagi-suk-mt-lite` (lite), both of which are encoder–decoder sequence-to-sequence models trained on the same 31,102 literary sentence pairs, split along whole document groups into 29,962 training, 993 validation, and 147 test pairs so that no evaluation sentence shares a source document with training data. Generation always begins from a forced Sukuma start token, and the quality SKU is simply the checkpoint that achieves the best held-out chrF2, while the lite SKU is a smaller sibling trained on the same table and recipe. On the document-group test set, the quality model reaches chrF2 **42.6** and BLEU **20.0**, while the lite model reaches chrF2 **34.8** and BLEU **12.7**. These numbers describe a research preview rather than a production-grade translation service, and the systems translate in one direction only: English is not accepted as a source language and must first be pivoted through Swahili.

1 Introduction

Sukuma (Kisukuma, ISO 639-3 `suk`) is a Bantu language spoken by several million people across the Lake Victoria region of northern Tanzania, making it one of the country's largest ethnic-language communities despite having very little presence in digital language tools. Like many Bantu languages of the region, Sukuma shares a broadly agglutinative grammar with extensive noun-class marking and verbal morphology, and it is written far less often than it is spoken. In everyday life, Sukuma speakers routinely rely on Swahili as the regional language of literacy and written communication, which makes Swahili–Sukuma parallel text the most practical entry point for building a translation system, since aligned Sukuma–English or Sukuma–French data is essentially unavailable at any scale. Public parallel data for the Swahili–Sukuma pair remains scarce even so, and the table used in this report should be understood as an early, literary-register foundation rather than a comprehensive corpus. Everyday conversation, news reporting, chat messages, and social media text are not represented in the NileAGI table, and this report makes no claim of coverage over those domains.

NileAGI's machine translation SKUs are built around a single, narrowly defined task: Swahili text goes in, Sukuma text comes out, rendered in a Latin orthography that preserves vowel macrons (`ā ē ī ō ū`). There is currently no public Sukuma-to-Swahili SKU and no public English-to-Sukuma SKU, so anyone who needs an English-language cascade should first translate English

into Swahili with any general-purpose system, then pass that Swahili text through one of the two models described here.

We publish two operating points that share the same underlying data and training recipe rather than representing two different tasks, as summarised in Table 1.

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 models live in the collection https://huggingface.co/collections/nileagi/nileagi-suk, and this document is concerned only with these two translators rather than with the wider NileAGI Sukuma toolchain.

The remainder of the report is organised as follows. Section 2 sets out the supported task, translation direction, and orthographic conventions. Section 3 describes the parallel table and the document-group splitting strategy used to prevent leakage. Section 4 covers the model architecture, the tokenizer extension work, the decoding constraint, and the optimisation setup. Section 5 reports validation and test metrics for both SKUs. Section 6 discusses qualitative translation behaviour beyond the headline numbers. Section 7 covers inference code, intended use, and licensing.

2 Task and language notes

2.1 Direction

The only supported mapping is Swahili → Sukuma. The public tokenizers expect the source language to be marked as `swh_Latn` and the target beginning-of-sequence token to be set to `suk_Latn`, and both settings must be applied at inference time. If either tag is omitted, the decoder will not reliably produce Sukuma output.

2.2 Script and spelling

Model output is written in Latin script, and NileAGI's convention is to keep vowel macrons throughout. Two written Sukuma practices coexist in real-world use, one with diacritics and one without, so models trained on this table will tend to look like they are experiencing domain shift when evaluated against macron-free text. The reverse also holds: users who strip macrons from the model's output are no longer matching the reference orthography that was used during evaluation, and any comparison against such stripped text should be treated with caution.

2.3 Length

On the NileAGI table, Sukuma text is consistently longer than its Swahili source: 733,283 target words against 587,871 source words overall, or roughly 1.25× as many whitespace tokens per sentence on average (23.6 Sukuma words versus 18.9 Swahili words per sentence pair). Because of

this systematic length difference, character-level metrics provide a more reliable primary score for this pair than word-level BLEU.

2.4 Register

The underlying table consists of literary written language, produced as read-aloud text rather than spontaneous dialogue. Greetings and short formulaic expressions are in-domain only to the extent that they happen to appear in that literary register, and the models should not be assumed to generalise well to registers that were never observed during training. Chat messages, legal drafting, clinical notes, and breaking news reporting all fall outside the scope of what this data can support.

2.5 English

English never appears as a source field anywhere in the training table. A user who wants to start from English text should treat Swahili as a required intermediate step, translating English to Swahili first and then Swahili to Sukuma. Any translation errors introduced during that first hop are carried forward unchanged into the Sukuma output, and this report does not attempt to score the combined two-stage cascade.

3 Data

All machine translation supervision comes from a single literary Swahili–Sukuma sentence table of 31,102 pairs, released under the Nwulite Obodo Open Data License (NOODL) 1.0. The table consists of text pairs only and is not redistributed together with the MT checkpoints themselves.

3.1 Inventory

Table 2 summarises the composition of the parallel table used to train both SKUs, including sentence counts, document-level structure, and word counts on each side of the language pair.

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

Each row of the training table carries the fields `src_text`, `tgt_text`, `src_lang` (always `sw`),

`tgt_lang` (always `suk`), `split`, `doc_id` (an anonymised document group identifier), and `unit_id` (identifying the unit's position inside its document).

3.2 Why document-group splits

Random sentence-level splits tend to leak information between training and evaluation, since neighbouring units drawn from the same document typically share names, recurring constructions, and discourse-level context. To avoid this, we hold out entire document groups rather than individual sentences, so that a sentence appearing in the test set never has a neighbour from the same group sitting in the training set. The trade-off is a fairly small test set: large enough to rank the two SKUs against each other and to quote a headline chrF2, but not large enough to support a fine-grained error analysis broken down by genre or topic.

Table 3 shows how the 31,102 pairs are distributed across the three document-group splits.

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 figure to quote when comparing these systems to others. Validation exists purely to choose a checkpoint and to sanity-check that training is progressing rather than collapsing. Because the held-out groups are whole documents rather than a large random sample, the reported scores would move somewhat if a different partition of document groups were chosen; that sensitivity is an expected property of a document-level split at this scale, not evidence of an unstable model.

4 Models and training

Both SKUs are encoder–decoder sequence-to-sequence translators, and training updates the full network rather than a lightweight adapter. The two SKUs differ in capacity and in a handful of optimisation details so that each one fits comfortably within the same overall recipe; they should be understood as two sizes of the same system rather than as models built for different tasks.

4.1 Decoding constraint

At every evaluation step, and in the public inference snippet shown later in this report, generation is started by forcing the Sukuma language token `suk_Latn` as the beginning-of-sequence token, while the source tokenizer's language is set to `swh_Latn`. This pairing is part of the public API surface for both models. Users who call `generate` without also passing `forced_bos_token_id` should not expect to reproduce the behaviour reported in Section 5.

4.2 Target-side vocabulary

Sukuma macrons and word stems are poorly covered by a generic multilingual tokenizer trained mainly on higher-resource languages. To address this, we fit a unigram SentencePiece model on the **training** Sukuma side only (vocabulary size 2,000, character coverage 1.0), and add every resulting piece that is missing from the existing tokenizer. This process added **1,426** new pieces for both SKUs, and the embedding matrix is resized accordingly to accommodate the expanded vocabulary. Where a dedicated Sukuma language token did not previously exist, its embedding is initialised by copying the Swahili language token's embedding, ensuring that the forced beginning-of-sequence token always has a well-defined vector to start decoding from. This tokenizer extension step is identical across the quality and lite SKUs.

4.3 Optimisation

The training objective is label-smoothed sequence-to-sequence cross-entropy, using Hugging Face `Seq2SeqTrainer` defaults except where noted below. Mixed precision uses FP16, and gradient checkpointing is enabled throughout training. The learning rate is fixed at 1×10^{-4}, with an effective batch size of 16 sequences. During evaluation, generation is capped at 128 new tokens with the Sukuma beginning-of-sequence token forced, and the trainer retains whichever checkpoint achieves the highest **validation chrF2**.

Table 4 lists the settings shared by both SKUs, while Table 5 lists the handful of settings that differ between them.

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 truncation length so that its larger network still fits comfortably within the shared training recipe. The lite SKU stops training after two epochs because validation chrF2 had already identified its best checkpoint by that point, and continuing further would not have changed which weights were shipped. Neither SKU uses the test split for early stopping or any other form of model selection.

5 Evaluation

5.1 Protocol

All headline numbers in this report come from Hugging Face's `Seq2SeqTrainer.evaluate`, run with `predict_with_generate`, forced target beginning-of-sequence tokens, and a maximum of 128 new tokens per generated sequence.

- **chrF2** (`evaluate` / `chrf`): a character n-gram metric with $\beta = 2$, scored on a 0–100 scale. This is treated as the **primary** metric throughout the report.
- **BLEU** (`evaluate` / `bleu`, default tokenizer): reported $\times 100$ to match commonly used tables. Treated as a secondary metric.
- **Eval loss**: token-level loss computed on teacher-forced labels, not on freely sampled generated text.

BLEU is a comparatively poor fit for this language pair. Sukuma is morphologically rich, runs about 1.25× longer than Swahili on this table, and decoding sometimes inserts spaces inside what should be single words. As a result, a system can look noticeably worse under BLEU than a human reader would judge it to be from the same generated string. We continue to report BLEU purely for comparability with other published MT tables, but we do not use it to rank the two SKUs against each other.

5.2 Validation

The validation set of 993 pairs is the selection set used to choose each shipped checkpoint. Table 6 shows chrF2 and BLEU after each training epoch, with the shipped checkpoint for each SKU marked in bold; because lite is shipped after epoch 2, it has no epoch-3 row to report.

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

The quality SKU keeps improving through all three epochs, with chrF2 rising from 39.1 to 44.6 and finally to 46.3. The lite SKU is shipped at epoch 2, where its chrF2 has risen from 34.0 to 39.1. At the selected checkpoints, the gap between the two SKUs on validation is roughly 7 chrF2 points (46.3 versus 39.1).

5.3 Test

Table 7 is the table to quote when citing this release. The test set consists of 147 pairs drawn from held-out document groups, with the primary metric, chrF2, marked in bold.

The quality SKU leads on every test metric: a 7.8-point chrF2 advantage, a 7.3-point BLEU advantage, and a lower evaluation loss. Lite remains the right choice when the quality checkpoint's

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

size is impractical for a given deployment, rather than when the single best-quality string is required.

The drop from validation to test scores (quality falling from 46.3 to 42.6 chrF2; lite falling from 39.1 to 34.8) is expected given the evaluation design. The test split is a handful of whole documents rather than a large random sample of sentences, so a one-point movement in chrF2 on this split should not be over-interpreted as a meaningful change in underlying model quality.

A chrF2 score in the mid-30s to low-40s range typically means the generated output carries the intended meaning reasonably well, but does not form a clean match against the human reference translation. This level of quality reflects a research preview rather than a substitute for a professional human translator.

6 Qualitative behaviour

Short Swahili greetings tend to land in plausible, natural-sounding Sukuma. On the quality SKU, for instance, `Habari yako?` decodes as `Ūlī mhola?`. Formulaic thanks and welcome expressions fall into the same band of behaviour: usable in practice, though never guaranteed to be correct.

Longer literary sentences are noticeably less stable than short greetings. Proper names and rare word stems are often approximated rather than carried through unchanged, and because the underlying table is drawn from literary writing, the models have not learned to handle conversational particles or code-switching with English or Swahili slang.

A recurring defect worth flagging is the appearance of **spaces inside Sukuma words**. These extra spaces inflate word-level mismatch scores and make BLEU look considerably worse than the character-level overlap captured by chrF2 would suggest. Any downstream display logic should therefore avoid treating every space in the output as a genuine morphological boundary, and when scoring new text against these models, chrF2 should be preferred over BLEU.

7 Using the models

7.1 Inference

Using either model requires setting the source language and forcing the Sukuma beginning-of-sequence token at generation time. The snippet in Listing 1 targets the quality SKU; substitute the repo id with `nileagi/nileagi-suk-mt-lite` to use the lite model instead.

Listing 1: Inference example (Python)

```
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer

```

```python
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))
```

All generation reported in this document used `max_new_tokens=128`. Longer documents should be split into individual sentences or short paragraphs before decoding, and there is currently no public English tokenizer path: English text should be translated into Swahili first.

7.2 Which SKU

Use the quality SKU unless the deployment's runtime environment cannot hold it. The lite SKU is trained on exactly the same pairs, with the same tokenizer extension and the same forced beginning-of-sequence behaviour, and it trails the quality SKU on every reported metric. Avoid mixing the two models within a single document if consistent spelling and spacing matter to the final output.

7.3 Intended use

- Literary Swahili → Sukuma translation.
- The final hop of an English → Swahili → Sukuma cascade.

7.4 Out of scope

- Sukuma → Swahili translation.
- Direct English → Sukuma translation.
- 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 to 40s is usable for research purposes, but it is not a replacement for a human translator.

2. **Register.** Spoken Sukuma and alternative orthographic conventions will present as domain shift to these models.
3. **Test size.** With 147 document-group pairs, the headline numbers are stable enough to prefer quality over lite, but not precise enough to claim an exact human-parity gap.
4. **Spacing.** Intra-word spaces in the output can hurt BLEU and confuse some downstream tokenizers.
5. **Pivot error.** Any English cascade carries forward whatever errors occur during the Swahili translation step.
6. **Rare stems.** Names and low-frequency word types are often approximated rather than translated exactly.

7.6 Licence

Table 8 summarises the licensing terms that apply to the model weights and to the underlying parallel training table separately, since the two are not covered by the same 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

In practice this means non-commercial use of the weights is permitted with attribution and share-alike, while commercial use of the weights requires a separate written agreement with NileAGI.

8 Conclusion

NileAGI releases two one-way Swahili → Sukuma translators, trained on a document-group split of 31,102 literary sentence pairs so that evaluation never leaks information from training documents. The number to quote is test chrF2: 42.6 for the quality SKU and 34.8 for the lite SKU. Choose the quality SKU whenever the accuracy of the generated string matters most, and choose the lite SKU whenever the deployment's footprint is the binding constraint. Both models require the forced beginning-of-sequence token `suk_Latn` together with the source language tag `swh_Latn` to behave as reported here. As with NileAGI's other Sukuma releases, these systems represent a research preview built on a literary, macron-based orthography rather than a general-purpose Sukuma translation product.

9 Citation

Listing 2: BibTeX entry

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

References

[1] M. Popović. *chrF: character n-gram F-score for automatic MT evaluation.* WMT, 2015. — the primary evaluation metric reported throughout Section 5.

[2] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu. *BLEU: a Method for Automatic Evaluation of Machine Translation.* ACL, 2002. — the secondary metric reported alongside chrF2 in Section 5.

[3] T. Kudo, J. Richardson. *SentencePiece: A Simple and Language Independent Subword Tokenizer and Detokenizer for Neural Text Processing.* EMNLP: System Demonstrations, 2018. — the unigram model used to extend the target-side vocabulary in Section 4.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. *Attention Is All You Need.* NeurIPS, 2017. — background on the encoder–decoder sequence-to-sequence architecture class used by both SKUs.

[5] T. Wolf et al. *Transformers: State-of-the-Art Natural Language Processing.* EMNLP: System Demonstrations, 2020. — the `transformers` library and `Seq2SeqTrainer` used for training and evaluation.

[6] NLLB Team, M. R. Costa-jussà et al. *No Language Left Behind: Scaling Human-Centered Machine Translation.* arXiv:2207.04672, 2022. — background on the `swh_Latn`/`suk_Latn` FLORES-200-style language-code convention used to tag source and target languages in Section 2.

[7] ∀, W. Nekoto et al. *Participatory Research for Low-resourced Machine Translation: A Case Study in African Languages.* Findings of ACL: EMNLP, 2020. — background on community-driven low-resource African MT efforts, the broader research context for a Swahili–Sukuma system.

[8] F. Sánchez-Martínez, V. M. Sánchez-Cartagena, J. A. Pérez-Ortiz, M. L. Forcada, M. Esplà-Gomis, A. Secker, S. Coleman, J. Wall. *An English–Swahili Parallel Corpus and its Use for Neural Machine Translation in the News Domain.* EAMT, 2020. — prior published work on building Swahili parallel corpora and MT systems, relevant background for Section 3.

[9] D. P. Kingma, J. Ba. *Adam: A Method for Stochastic Optimization.* ICLR, 2015. — the optimizer family underlying the Hugging Face training defaults used here.

[10] P. Micikevicius et al. *Mixed Precision Training.* ICLR, 2018. — background on the FP16 training setup described in Section 4.