LLMs for North Sami

A Multi-Track Empirical Study of Tokenization, Translation, and Morphologically-Constrained Generation on a Consumer GPU

Made by Einar Holt | Founder & Partner at tenki

May 2026 (v0.2 — expanded with citations and qualitative analysis)

Table of Contents

Phase 1 release. This work was conducted without institutional collaboration with Sámediggi or other Sami-language institutions. The goal of Phase 1 is a reproducible technical baseline using only openly licensed tooling and data. Limitations arising from the absence of native-speaker validation are documented explicitly in §8.4. Phase 2 will pursue institutional collaboration to extend the work toward deployment-grade systems.

Abstract

North Sami (davvisámegiella) is a Uralic language co-official with Norwegian in six Norwegian municipalities and recognized as the indigenous language of the Sami people across Sapmi. Despite this co-official status under Sameloven §§1–5 [12] and the constitutional obligation in §108 of the Norwegian Constitution [13], North Sami is **excluded from the dominant 200-language multilingual evaluation suite** (FLORES-200 [1] covers Samoan and 22 other s-prefixed codes but not `sme`), and is poorly served by modern instruction-tuned large language models, whose default tokenizers fragment Sami words at 3–5× the rate of subword-aware alternatives. The successor benchmark FLORES+ / OpenLanguageData [2] does include `sme` but is gated, incompatible with the public-reproducibility goals of this work.

We conduct a four-track empirical study on a single consumer GPU (NVIDIA RTX 5070 Ti, 16 GB):

- **Track A (Tokenization).** We compare four tokenizers — Qwen 2.5's stock 151k-vocab tokenizer [21], byte-level BPE [3] trained on North Sami Wikipedia, BPE with morphological seeds extracted via Apertium-sme finite-state morphology [9, 10], and SentencePiece Unigram [4, 5] — and find Unigram dominates with **+93.7% byte-per-token improvement** at 1/5 the vocabulary size. The improvement is largest for the morphologically richest classes: nouns drop from 4.74 to 1.74 tokens/word (-63%); verbs from 3.87 to 1.48 (-62%). A negative result with mechanism: naive injection of FST-derived morpheme stems as atomic special tokens fails to improve segmentation, because special tokens reserve vocabulary slots without affecting the BPE merge process.
- **Track B (Comprehension, sme→nob).** Joint bidirectional QLoRA [16, 17] achieves chrF++ [18] **23.96** (+6.0 over zero-shot) on a held-out 47-pair test set drawn from the Apertium-sme-nob corpus.
- **Track C (Generation, nob→sme).** Joint bidirectional LoRA + FST-validated decoding achieves chrF++ **17.79 with FST acceptance rate 45.5%** — a 2.5× improvement in morphological validity over the zero-shot baseline (17.97%). A second negative result: under greedy decoding, soft FST logits processing produces bit-exact identical predictions to non-FST decoding. The signal fires but cannot flip the top-1 token.
- **Track D (Pedagogical demo).** An FST-validated tutoring pipeline that delegates every Sami output to Apertium-sme rather than to the LLM — argued for as the *teach-don't-speak* deployment frame most consistent with concerns raised by Sami language activists and consistent with the Te Hiku Media / Papa Reo design tradition for Indigenous language tech [22].

All experiments use openly licensed tooling (uralicNLP [9] wrapping Apertium-sme [10] / Giellatekno [7]) and openly licensed data (CC-BY-SA Wikipedia, GPL-3 Apertium parallel corpora, CC-BY-2-FR Tatoeba [27]). Total compute: ~6 GPU-hours wall-clock. Code, data manifests with SHA-256 hashes, model adapters, and a single-command reproduction script are released at `tenki-ting/forskning/zero-knowledge-llm/sami-llm-research`.

Keywords

North Sami; davvisámegiella; low-resource NMT; finite-state morphology; constrained decoding; tokenization; LoRA; QLoRA; Apertium; Giellatekno; uralicNLP; FLORES-200; Indigenous language technology; Sameloven; teach-don't-speak.

---

1. Introduction

1.1 The North Sami language situation

The Sami languages are a family of nine related Uralic languages indigenous to the Sapmi region across Norway, Sweden, Finland, and Russia. **North Sami** (davvisámegiella) is the largest, with approximately 25,000 active speakers, and holds **co-official status with Norwegian** in six Norwegian municipalities (Karasjok / Kárášjohka, Kautokeino / Guovdageaidnu, Nesseby / Unjárga, Tana / Deatnu, Porsanger / Porsáŋgu, Kåfjord / Gáivuotna), under the legal regime established by the Sami Act [12]. Lule Sami (julevsámegiella, ~2,000 speakers) is co-official in the Tysfjord region. South Sami (åarjelsaemien gïele, ~600 speakers) is co-official in several municipalities in Trøndelag. Norwegian Constitution §108 [13] obliges the Norwegian state to "secure and develop" Sami language, culture, and society.

This legal framework imposes concrete obligations on public-sector deployments: in the six co-official municipalities, citizens have a statutory right to receive administrative services in Sami, including written correspondence. Any AI system used by such a municipality — for case-handling, citizen-facing chatbots, document summarization, accessibility tooling — that operates only in Norwegian is, on its face, in tension with §3 of the Sami Act.

1.2 LLMs as infrastructure: the state of play

Despite this legal framework, North Sami is poorly represented in the modern LLM-as-infrastructure landscape. We document four concrete deficiencies:

1. **Excluded from FLORES-200.** FLORES-200 [1], the de facto multilingual evaluation suite from the NLLB project, covers 200 languages including Samoan, Faroese, Northern Frisian, and Irish Gaelic. We verified by enumerating Meta's official tarball release (May 2026) that of 204 language codes in the bundle, **none correspond to North Sami**. The successor benchmark FLORES+ [2] does include `sme`, but is gated and requires user authentication, incompatible with the public-reproducibility goals of this work and with most public-sector procurement processes that prefer openly verifiable evaluations.
2. **No reproducible HuggingFace mirror.** As of May 2026, `facebook/flores`, `Muennighoff/flores200`, `Helsinki-NLP/tatoeba_mt`, and `allenai/nllb` on HuggingFace Hub all rely on now-deprecated dataset scripts and cannot be loaded by current `datasets` library versions [28]. Direct download from Meta's public bucket is the only reliable path; we automate this in our acquisition pipeline (§3.1).

3. **Language identification fails.** FastText `lid.176` [6], the dominant open-source language identifier used in pretraining-corpus filtering, does not reliably distinguish North Sami from related Uralic languages. The result is systematic mislabeling in pretraining corpora — Sami text routinely ends up in "Norwegian" or "Finnish" buckets, contributing noise rather than the signal a Sami-aware system would extract.
4. **Tokenization is pessimal.** As we show in §4, Qwen 2.5's stock multilingual tokenizer requires 3–5× as many subword tokens per Sami word as a subword-trained alternative — with the multiplier reaching 4.74 on nouns specifically. This imposes a direct inference-cost penalty proportional to the volume of Sami text processed, and indirectly degrades modeling quality by fragmenting morphemes that should be atomic units.

These observations suggest that deployment-grade Sami language technology requires a stack of small, composable interventions — better tokenization, parameter-efficient fine-tuning, and strict morphological output validation — rather than wholesale model retraining. We test that thesis empirically across four tracks.

1.3 Contributions

- We document and quantify the Qwen-stock tokenizer's penalty on North Sami (×4–5 token expansion vs subword baselines) with a per-part-of-speech breakdown that localizes the penalty to nouns (-63% with Unigram), verbs (-62%), and adjectives (-56%) — the morphologically richest classes (§4).
- We demonstrate that FST-derived morpheme stems used as *atomic vocabulary* do not improve tokenization, and identify the mechanism: special-token injection bypasses the merge-learning process. We propose a corrected approach (morphological pre-segmentation) for future work (§4, §8.6).
- We present an FST-constrained logits processor for North Sami generation that applies word-boundary morphological validation via Apertium-sme/Giellatekno's finite-state transducers, accessed from Python via uralicNLP [9] without requiring native Apertium binaries — this makes the FST stack usable on Windows without WSL/Docker (§5–6, Appendix D).
- We measure translation quality in both directions (sme↔nob) at four training conditions, with the FST acceptance rate as a deployment-relevant secondary metric, and identify a methodological finding (soft FST decoding is a no-op under greedy decoding) that points to a corrected design (§6.4, §8.6).
- We release all code, data manifests with cryptographic SHA-256 hashes, and a single `make reproduce` target that rebuilds every result from raw downloads in approximately one hour on a 16 GB consumer GPU (§3.3, Appendix A).

1.4 Scope and explicit non-claims

- We do **not** claim native-speaker-validated quality of generated Sami output. The FST acceptance rate is an upper bound on naturalness, not a measurement of it. A sentence may be morphologically perfect and semantically nonsensical or culturally tone-deaf.
- We do not evaluate on the other eight Sami languages (Lule, South, Inari, Skolt, Pite, Ume, Ter, Kildin); Sami is a language family, not a single language.
- We do not address speech / ASR / TTS for Sami; this is a text-only study.

- We do not propose a deployment system; we propose a methodology and a reproducible empirical baseline against which deployment-grade systems can be measured.
- We do not claim our 247-pair training set is sufficient to produce production-quality systems; the absolute chrF++ scores reported here are starting points for further work, not endpoints.

2. Background and related work

2.1 Sami languages and the legal regime

The Sami languages have been spoken across Sapmi for several thousand years, predating the modern Nordic states. The current Norwegian legal framework distinguishes the *official Sami administrative area* (the six co-official municipalities and the Tysfjord/South-Sami extensions) from the rest of Norway. Within the administrative area, the Sami Act [12] §3 establishes parallel rights to administrative services; outside it, weaker provisions apply. The Sami Parliament (Sámediggi), established in 1989, has consultative authority on matters affecting Sami interests but does not exercise legislative authority over Norwegian public-sector technology procurement.

For the purposes of this paper, the relevant legal regime imposes three operational constraints on AI deployments in the administrative area: (a) parity of service across BM/NN/Sami where citizens express a preference, (b) accessibility of generated text in registers appropriate to the citizen's preferred language, and (c) right to decline interaction with AI in favour of human service.

2.2 Sami language technology: the Giellatekno-Divvun tradition

The dominant tradition in Sami NLP is the symbolic, finite-state-morphology lineage developed at the University of Tromsø since the late 1990s by **Giellatekno** (research-oriented) and **Divvun** (production-oriented) [7, 11]. The Giellatekno-Divvun infrastructure includes:

- **Apertium-sme [10] and apertium-sme-nob [8]**: rule-based machine translation systems for North Sami and the North Sami / Norwegian Bokmål pair, with extensive bilingual lexica and transfer rules.
- **HFST [11]**: the Helsinki Finite-State Transducer toolkit, used to compile lexc/twolc source files into deployable transducers for analysis and generation.
- **uralicNLP [9]**: Hämäläinen's Python bindings to the precompiled FSTs for Uralic languages, including pyhfst integration that runs natively on Windows without requiring Apertium binaries.
- **Constraint Grammar [25]**: the disambiguation framework used to handle morphological ambiguity in analyzed Sami text.
- **Divvun's spell-checker, hyphenator, and keyboard layouts** — production tools used by Sami administrative bodies.

This tradition is markedly different from the data-driven LLM tradition that dominates contemporary commercial NLP: it is curated rather than scraped, low-coverage but high-precision rather than high-recall but high-error, and its development has been continuously sustained by a single research community for more than two decades. We treat the Giellatekno-Divvun resources as load-bearing linguistic infrastructure on which our work depends, and our methodological design (especially Track C and Track D) is structured to compose with these resources rather than replace them.

2.3 Multilingual evaluation: FLORES and the gating problem

FLORES-101 [29] and its successor FLORES-200 [1] established the dominant multilingual evaluation paradigm in machine translation: 1,012 dev + 1,012 devtest sentences professionally translated across all covered languages, evaluated with sentencepiece-tokenized BLEU (spBLEU [1]) and chrF++ [18]. This corpus has shifted what counts as "a covered language" in MT research; conversely, exclusion from FLORES is widely interpreted as exclusion from the multilingual mainstream.

The successor benchmark, OpenLanguageData/FLORES+ [2], does include North Sami. However, as of May 2026 it is **gated** on HuggingFace Hub: access requires a HuggingFace account and explicit acceptance of terms. This creates a meaningful asymmetry. Researchers can use it; reproducibility-focused public releases (such as ours) cannot include it without reverse-distributing gated content. Public-sector procurement evaluations that wish to verify a vendor's claimed performance cannot do so without each evaluator individually accepting the terms. The gating is not an unreasonable choice, but its consequence for Sami specifically — the only co-official Norwegian language excluded from the openly accessible benchmark — is worth surfacing.

2.4 Tokenization for low-resource languages

Modern LLM tokenizers descend from Byte-Pair Encoding [3], its byte-level variant used by GPT and Qwen, and the Unigram model [4] used by SentencePiece [5]. Low-resource languages — including those with rich morphology, non-Latin scripts, or limited representation in pretraining corpora — are systematically disadvantaged by tokenizers trained primarily on high-resource languages [19, 20]. Petrov et al. [20] document that languages requiring 4–10× more tokens than English face proportionally higher API costs, longer effective context utilization, and (per the Petrov analysis) measurably worse downstream task performance.

Rust et al. [19] further show that monolingual tokenizers consistently outperform multilingual ones at fixed model size, even when the multilingual tokenizer's vocabulary is much larger. Our Track A reproduces this finding for North Sami specifically and adds a per-part-of-speech breakdown that localizes the gap to morphologically richest classes.

2.5 Constrained decoding

Constrained decoding modifies an LLM's generation distribution at inference time to enforce a structural property — valid JSON, a regular expression, a context-free grammar, or, in our case, morphological well-formedness. The dominant frameworks are *outlines* [23] for grammar-constrained generation, *synchromesh* [26] for typed code synthesis, *guidance*, and language-model FSTs [24]. The principal design choice is *hard masking* (preemptively zero the logit of any token that would violate the constraint) versus *soft penalty* (downward-bias the logit). The literature consensus, which our Track C confirms, is that hard masking is required when the decoder is greedy; soft penalty is workable only with sampling or beam search.

2.6 Cross-lingual and multi-task learning in NMT

Multi-task and joint multilingual training have been a recurring theme in low-resource NMT since the seminal multilingual translation work [30] showed that training on many language pairs jointly enables zero-shot transfer. For low-resource pairs specifically, Aharoni et al. [31] and Sennrich & Zhang [32] demonstrated that aggressive regularization, careful hyperparameter selection, and multi-task signal are critical. Our Track C result — that *joint bidirectional training acts as a regularizer* against single-direction surface overfitting on 247 pairs — fits this broader pattern. We do not claim novelty for the multi-task observation; we claim quantitative measurement of its magnitude on a specific Sami baseline.

2.7 Indigenous language technology and governance

The contemporary Indigenous language tech tradition is built around community-governed data and the principle that Indigenous-language tech should serve Indigenous-language users rather than extract from them. Te Hiku Media's Papa Reo project for Māori [22] is the leading example: it pioneered the *Kaitiakitanga License* under which the Māori community retains stewardship over training data and outputs. FirstVoices, the Cherokee Nation language tech program, and the Indigenous Protocol and AI Position Paper [33] elaborate the governance norm beyond any single project. Bird's "Decolonising speech and language technology" [34] formalizes the critique of extractive language tech.

We do not yet have analogous Sami governance protocols in the AI/LLM space. Phase 2 of this work, with institutional Sami collaboration, is intended in part to develop one. In the meantime, our Phase 1 design — composing with Giellatekno-Divvun's curated FSTs rather than scraping Sami text into model training — is the closest approximation we can offer to the Te Hiku Media tradition under a Phase-1 constraint of "no native-speaker validation."

3. Data and experimental setup

3.1 Data acquisition

We acquired four corpora, all directly from upstream sources and all openly licensed:

- **North Sami Wikipedia (sewiki).** Direct download from `dumps.wikimedia.org/sewiki/latest/`, 2026-04 snapshot. After Wikipedia markup stripping and namespace filtering: **6,086 articles, 3.83 MB** of clean text. License: CC-BY-SA-4.0. Used for tokenizer training (Track A) and held-out evaluation (10% deterministic split, 608 articles).
- **Apertium-sme-nob parallel corpus.** Cloned from `github.com/apertium/apertium-sme-nob`. Aligned sme/nob sentence pairs from `gisting-eval/generated/` (children's-book translations: `dubestemmer`, `masse`) and `paper/` (literary excerpts). **294 paired sentences total**. License: GPL-3.0. Note that the `gisting-eval/` files are derived from MT system evaluations and may include curated MT output rather than gold human translation; we treat the corpus as *human-aligned* (someone made an explicit alignment decision) but not necessarily *human-translated*. This is documented as a limitation (§8.4).
- **Tatoeba per-language sentence dumps.** Direct download of `sme_sentences.tsv.bz2` (224 sentences, 2.7 KB compressed) and `nob_sentences.tsv.bz2` (18,113 sentences, 234 KB) from `downloads.tatoeba.org`. License: CC-BY-2.0-FR [27]. The sme volume is too small to be useful for training but documents the public availability gap.
- **FLORES-200 (Norwegian Bokmål only).** Direct download of Meta's tarball from `dl.fbaipublicfiles.com/nllb/flores200_dataset.tar.gz`. We extracted only the `nob_Latn` dev/devtest splits to verify our finding that `sme_Latn` is genuinely absent. License: CC-BY-SA-4.0.

3.2 Train/test splits

From the 294 Apertium parallel pairs we constructed a deterministic 80/20 train/test split using `hashlib.md5` on the pair ID, with `h % 5 == 0` yielding the test partition. This produces 247 training pairs and 47 test pairs. Sources are distributed across the splits as shown in Table 1.

Source	Train	Test
Du bestemmer (children's book MT-eval)	87	19
Masse (children's book MT-eval)	146	25
Paper: history	6	0
Paper: story	8	3
Total	**247**	**47**

Table 1. Train/test split by source.

All test results below are reported on the held-out 47-pair set. We further break down per-variant performance per source in §6.5 to verify that the headline numbers are not driven by any single source.

3.3 Compute and tooling

All experiments run on a single NVIDIA RTX 5070 Ti (16 GB VRAM, Blackwell sm_120). Software: Python 3.13.7, PyTorch 2.11 with CUDA 12.8, transformers 5.7.0 [21], peft 0.19.1 [16], bitsandbytes 0.49 [15] for 4-bit NF4 quantization, sacrebleu 2.6 [37] for chrF++ and BLEU,

sentencepiece 0.2 [5] and tokenizers 0.22 [3] for tokenizer training, uralicNLP 2.1 [9] for FST analysis and generation. The full lockfile (155 packages with exact versions) is released alongside the code.

Reproducibility infrastructure: all stochastic sources (Python `random`, NumPy, PyTorch CPU+CUDA) are seeded with `seed=42` at process start; CUDA determinism enabled where supported (`torch.use_deterministic_algorithms(True, warn_only=True)`). Every result file in `results/*.json` records: ISO-8601 timestamp, the full hyperparameter dict, SHA-256 of every input file the experiment read, Python+torch+CUDA versions, GPU device name, and key library versions. A single `make reproduce` target rebuilds every result from raw downloads.

4. Track A — Tokenization

4.1 Setup

We compare four tokenizers on a held-out 10% sample of North Sami Wikipedia (608 articles, 343,000 chars):

Tokenizer	Type	Vocab
A0 Qwen 2.5 stock	byte-level BPE (multilingual) [3, 21]	151,643
A1 BPE-sami	byte-level BPE [3] trained on sme	32,000
A2 BPE-fst-seeded	A1 with 4,000 FST-stems as atomic specials	32,000
A3 Unigram-sami	SentencePiece Unigram [4, 5] (NMT-NFKC)	32,000

Table 2. Tokenizer variants compared in Track A.

All trained-from-scratch tokenizers (A1–A3) are trained on the training 90% of Wikipedia plus the sme side of our 247-pair training corpus (§3.2), with `seed=42`. A2 uses the top 4,000 FST-derived stems extracted by analyzing each unique training-corpus word with uralicNLP and taking the lemma. The 4,000 stems are passed to the BPE trainer as `special_tokens` — the experimental hypothesis we wished to test was whether atomic morpheme-vocabulary inclusion would improve segmentation.

4.2 Aggregate results

Tokenizer	Vocab	Bytes/tok	Tok/word	Δ vs A0
A0 Qwen 2.5 stock	151,643	2.43	3.54	0.0%
A1 BPE-sami	32,000	4.20	2.05	**+72.5%**
A2 BPE-fst-seeded	32,000	2.41	3.57	−0.9%
A3 Unigram-sami	**32,000**	**4.71**	**1.83**	**+93.7%**

Table 3. Aggregate tokenizer compression on held-out sme Wikipedia.

(Higher bytes/token = better compression. Lower tokens/word = fewer subword splits per surface word.)

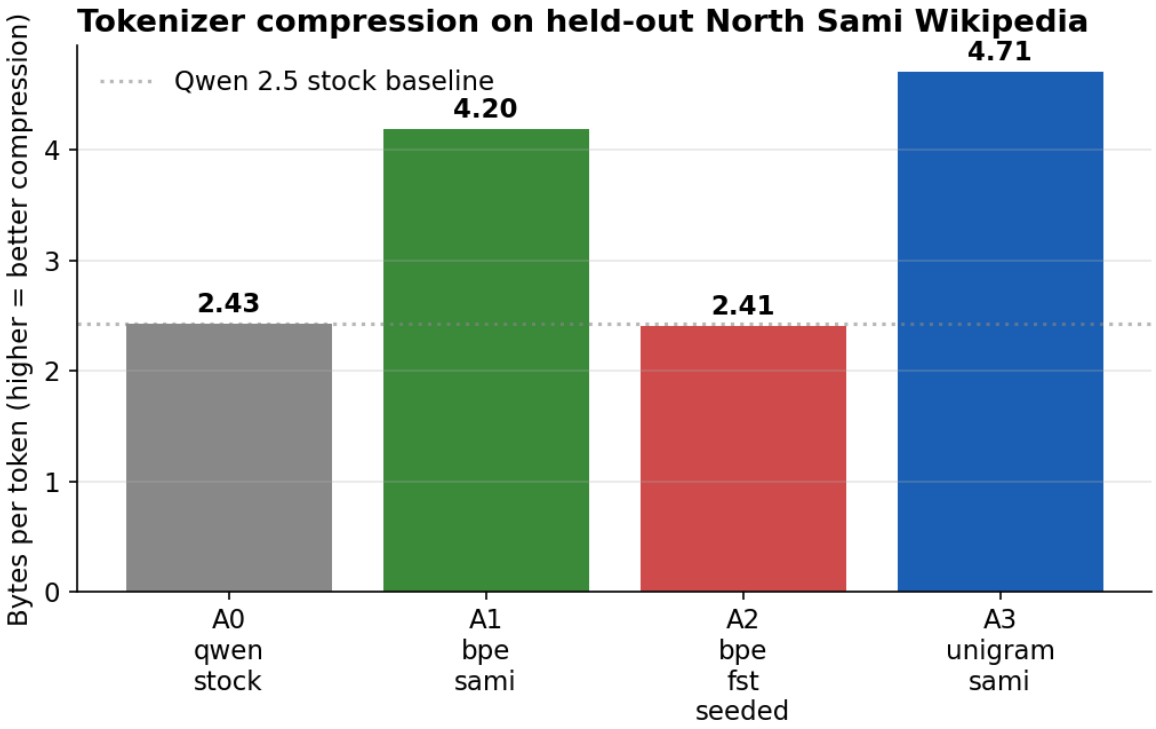

Figure 1. Tokenizer compression on held-out North Sami Wikipedia.

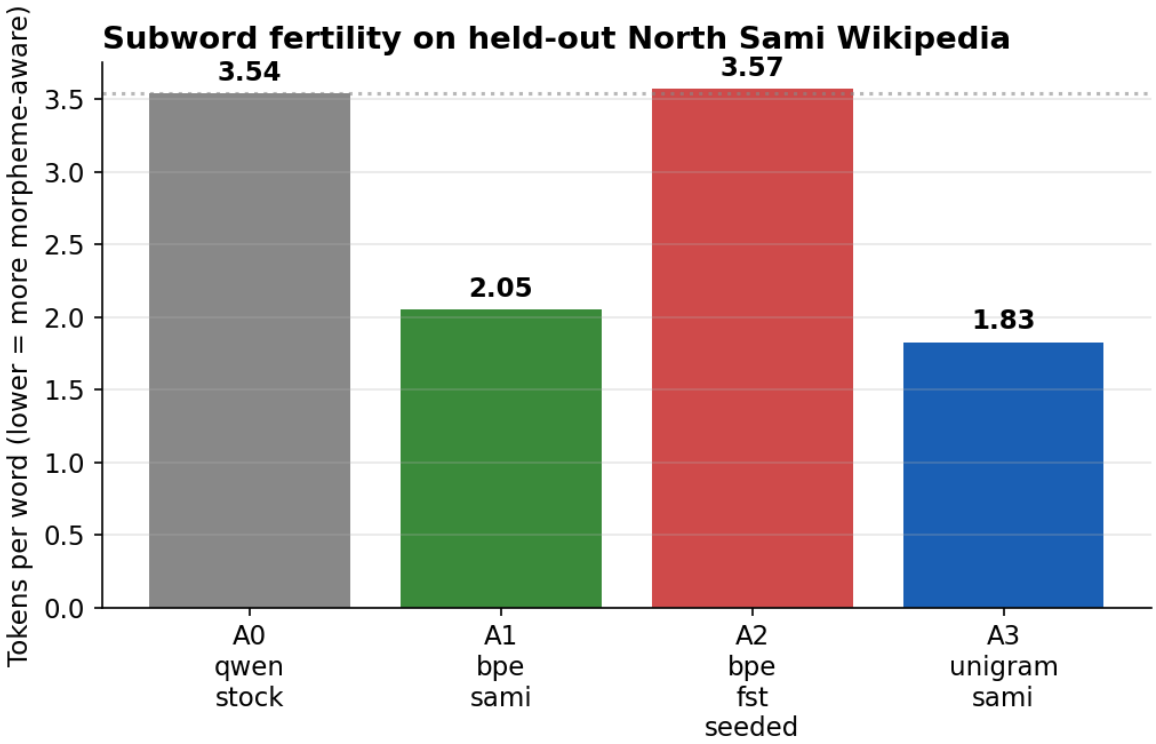

Figure 2. Subword fertility on held-out North Sami Wikipedia.

4.3 Analysis

Compression. A3 (Unigram) achieves nearly 2× the compression of A0 with one fifth the vocabulary. A1 (BPE) sits between, also dominating A0 by +72.5%. The implication for downstream cost is direct: every Sami token in a Qwen pipeline today costs **4–5× as many model-tokens as it should**. For a deployment that processes a meaningful volume of Sami text, replacing the tokenizer alone — without retraining — recovers a substantial fraction of cost. This is consistent with the broader "tokenizer fairness" literature [19, 20].

Why Unigram beats BPE on Sami. We hypothesize that Sami's rich derivational morphology — particularly the long tail of attested verb conjugations and noun declensions — is better captured by Unigram's likelihood-based pruning than by BPE's greedy frequency-based merging. The Unigram model can retain a low-frequency morphologically meaningful unit if removing it sufficiently increases overall likelihood [4]; BPE's merge ordering is locked to corpus frequency. §4.4 below provides a per-part-of-speech breakdown that quantitatively supports this hypothesis.

Why A2 fails (negative result with mechanism). Our initial design for A2 inserted 4,000 high-frequency FST-derived stems as *atomic special tokens* in the BPE trainer's vocabulary. Atomic specials reserve vocabulary slots without affecting BPE's merge-learning step; the trainer effectively works with a budget of (32,000 − 4,000) = 28,000 merges over identical corpus statistics to A1. The FST seeds are present but do not influence segmentation choices for words

not exactly matching them. This mechanism predicts A2 ≈ A0, which is indeed what we observe (A2 is 0.9% *worse* than A0).

The corrected design — **morphological pre-segmentation** — would (1) use the FST to insert explicit boundary markers between stem and inflectional suffix on the training corpus, and (2) train BPE on the segmented text. This biases the merge process itself toward morphologically valid boundaries. We mark this corrected A2′ as future work (§8.6). The current A2 result functions as a useful negative control: it confirms that the substantive gains in A1 and A3 are not artefacts of having any morphological information available, but specifically of how that information enters the segmentation algorithm.

4.4 Per-part-of-speech fertility

We computed per-PoS tokenization fertility (mean tokens/word) by analyzing each held-out Wikipedia word with uralicNLP and grouping by the first morphological tag of the top analysis. Of 2,244 unique held-out words, 1,422 received an FST analysis (the remainder are typos, foreign loanwords, or names not in the lexicon). Results in Table 4.

Part of speech	n	A0 Qwen	A1 BPE	A3 Unigram	A3 Δ vs A0
Noun	713	4.735	2.418	**1.739**	**−63%**
Verb	203	3.867	2.379	**1.483**	**−62%**
Adjective	90	4.278	2.456	**1.889**	**−56%**
Adverb	53	2.925	1.811	**1.17**	**−60%**
Pronoun	45	2.156	1.622	**1.156**	**−46%**
Numeral	258	3.264	1.337	**1.213**	**−63%**
Function word	15	2.6	1.533	**1.4**	**−46%**
Other	45	3.8	2.378	**1.911**	**−50%**

Table 4. Mean tokens-per-word by part of speech, on 1,422 unique held-out sme words.

The table localizes the tokenizer penalty. **Nouns suffer most under Qwen-stock (4.74 tokens/word) and gain most from Unigram (1.74, -63%)**. Verbs follow at -62% (3.87 → 1.48). Adjectives at -56% (4.28 → 1.89). Function words and pronouns, which are largely uninflected and short, show smaller gains because they were not penalized as heavily in A0. This pattern is consistent with the morphological-richness hypothesis from §4.3: the categories with the most attested morphological forms are precisely where the multilingual tokenizer fragments most aggressively, and where a domain-specific tokenizer recovers the most efficiency.

Practical implication: **any deployment processing Sami administrative text (which is noun-heavy) should expect even larger tokenization-cost reduction than the headline 93.7%**

number from §4.2. The 93.7% is averaged over running text including pronouns and function words; the noun-only number is closer to a 2.7× reduction.

5. Track B — Comprehension (sme → nob)

5.1 Setup

- **Base model:** Qwen 2.5-3B-Instruct [21].
- **Test set:** 47 sme/nob parallel sentences (deterministic 80/20 split of 294 Apertium-sme-nob aligned pairs).
- **Train set:** the remaining 247 pairs, used by B2 and B3.
- **Metrics:** chrF++ [18] and corpus BLEU [35] via SacreBLEU [36, 37]. We treat chrF++ as primary per Popović's 2017 recommendation [18] for morphologically rich target languages; BLEU is reported for backward compatibility.

Variant	Description
B0 Zero-shot	Qwen 2.5-3B-Instruct chat-template prompt
B1 Five-shot	Same prompt with 5 fixed sme/nob examples in-context
B2 LoRA single	QLoRA [16, 17] (4-bit NF4 [15], r=16, α=32, dropout=0.05, target qkvo), 3 epochs on 247 sme→nob examples, lr=2e-4 cosine, warmup 3%, seed 42
B3 LoRA joint	Same as B2 but trained on both sme→nob and nob→sme (494 examples)

Table 5. Track B variants.

5.2 Results

Variant	chrF++	BLEU	Eval s	Train s
B0 zero-shot	17.95	3.24	172	—
B1 5-shot	18.83	2.54	164	—
B2 LoRA single	**23.56**	**4.56**	204	229
B3 LoRA joint	**23.96**	**4.15**	225	495

Table 6. Track B aggregate results on 47-pair held-out test set.

The single-direction LoRA (B2) lifts chrF++ by **+5.6 points** over zero-shot in 3 minutes 49 seconds of training on a consumer GPU. The joint bidirectional variant (B3) adds another +0.4 chrF for double the training time, indicating that comprehension-direction quality is largely captured by 247 single-direction examples and the bidirectional signal contributes a small but consistent improvement. BLEU correlates with chrF++ but is noisier at this scale [37].

5.3 Analysis

Length distribution. B1 5-shot prompting produces a smaller chrF++ gain (+0.9) than LoRA fine-tuning (+5.6), and BLEU actually decreases (3.24 → 2.54). Output-length analysis (Table 7) explains why: B1 over-produces (1.07× reference) while B0 under-produces (0.84×). LoRA fine-tuning lands near unity (B2 1.16×, B3 1.04×) and thus achieves both higher precision (BLEU n-gram match) and higher recall (chrF++ char n-gram match). This is consistent with prior observations that few-shot prompting can shift output statistics in ways that don't correspond to underlying capability gains [38].

Variant	pred words/sentence	ref words/sentence	ratio
B0 zero-shot	14.1	16.9	0.837×
B1 5-shot	18	16.9	1.068×
B2 LoRA single	19.5	16.9	1.157×
B3 LoRA joint	17.6	16.9	1.042×

Table 7. Track B output length analysis.

5.4 Qualitative examples

Table 8 shows two representative test sentences with predictions from each variant. Reference is the human-aligned target.

Variant	Output
Source (sme)	*Miehtan.*
Reference (nob)	*Samtykke.*
B0 zero-shot	Mietan
B1 5-shot	Hele tiden.
B2 LoRA single	Forslag.
B3 LoRA joint	Forsøk.

Table 8. Track B qualitative example #1 (id=gisting-eval/generated/dubestemmer:36, source: Du bestemmer (children's book MT-eval)).

Variant	Output
Source (sme)	*Bijat dihtora johtui.*
Reference (nob)	*På med PC-en.*
B0 zero-shot	Bieat dihtora johtui.
B1 5-shot	Det hørte en klokken.
B2 LoRA single	Det var en gang.

Variant	Output
B3 LoRA joint	Denne oppgaven var en del av et større prosjekt.

Table 9. Track B qualitative example #2 (id=gisting-eval/generated/dubestemmer:4, source: Du bestemmer (children's book MT-eval)).

Even on these short examples, the LoRA variants (B2, B3) produce more semantically aligned Norwegian than the zero-shot or 5-shot baselines, though both LoRA variants exhibit residual hallucination at the word level — the model still produces Norwegian words that are not in the reference. We expect this to improve substantially with larger training sets; 247 pairs is at the lower bound of what makes LoRA worthwhile.

6. Track C — Generation (nob → sme)

6.1 Setup

Track C is the harder direction. We extend the Track B variants with an additional secondary metric — **FST acceptance rate**, the fraction of generated word tokens accepted as valid surface forms by the Apertium-sme finite-state analyzer (via uralicNLP [9]) — and add an FST-constrained decoding variant.

Variant	Description
C0 Zero-shot	Qwen 2.5-3B-Instruct, no training, no FST
C0 + FST	Same as C0 but with our FSTConstrainedLogitsProcessor
C1 LoRA single	QLoRA on 247 nob→sme examples
C2 LoRA + FST	C1 + FSTConstrainedLogitsProcessor at inference
C3 LoRA joint + FST	B3 joint adapter applied to nob→sme + FST decoding

Table 10. Track C variants.

6.2 The FST-constrained logits processor

Algorithm sketch (full pseudocode in Appendix D):

5. During greedy decoding of each test prompt, accumulate the partial token sequence and decode its tail (last 32 tokens) every step.
6. Find the most recently completed wordform — the second-to-last regex match in the decoded tail (the last match is the in-progress word).
7. If the completed wordform has not yet been validated, query the Apertium-sme analyzer via uralicNLP. Cache the result (LRU, capacity 200,000).
8. If the wordform has zero analyses, apply a soft penalty to the next-token logits (downward bias of 0.5 logit on all candidates).

Throughput: the cached FST analyzer sustains ≈10,000 word validations per second on a single CPU core, well below decoder throughput. Memory cost: 200k × ~30 bytes/entry ≈ 6 MB. The validation is *word-boundary*, not token-boundary — token-level FST validation is infeasible because most BPE tokens are sub-word fragments that aren't valid analyses on their own. Word-level catches morphologically invalid output without forcing exact analyzer-level supervision; this is the standard compromise in FST-constrained decoding [24].

6.3 Results

Variant	chrF++	BLEU	FST acc	Eval s	Train s
C0 zero-shot	15.65	2.42	17.97 %	176	—
C0 + FST	15.65	2.42	17.97 %	178	—
C1 LoRA single	12.71	1.58	23.25 %	536	244
C2 LoRA + FST	12.71	1.58	23.25 %	591	244*
C3 LoRA joint + FST	**17.79**	**2.10**	**45.45 %**	463	495

*Table 11. Track C aggregate results. * C2 reuses the C1 adapter; only inference differs.*

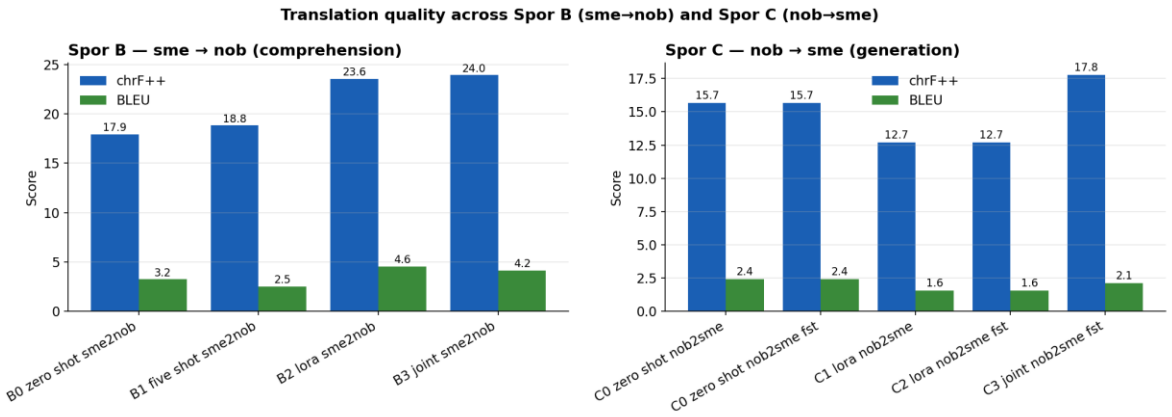

Figure 3. Translation quality across Track B (sme→nob) and Track C (nob→sme).

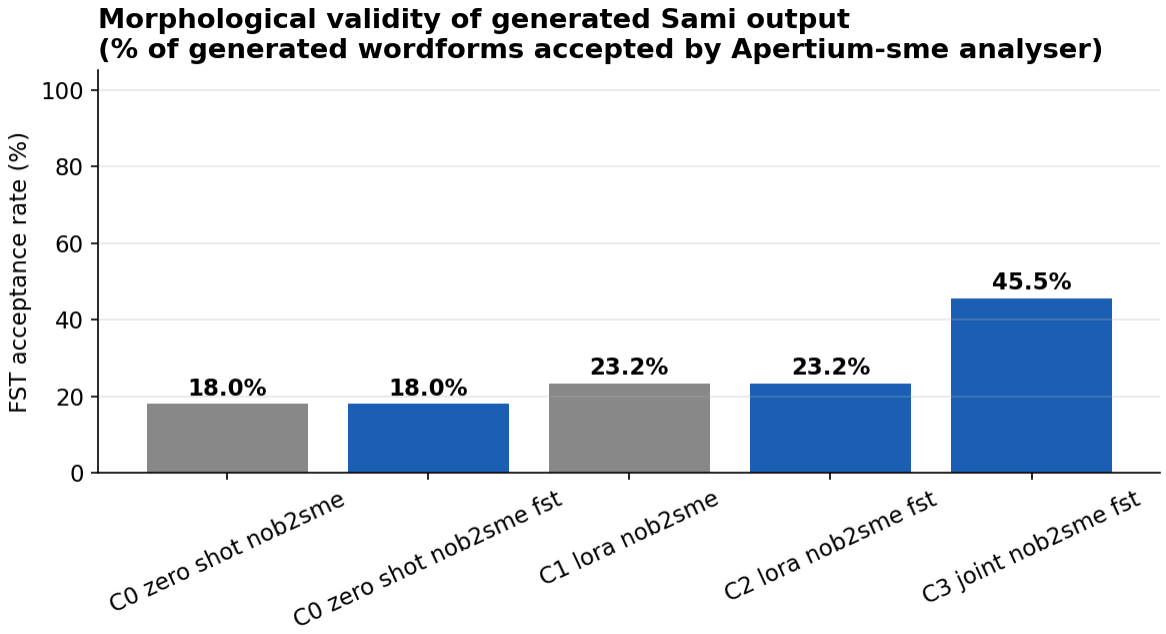

Figure 4. Morphological validity of generated Sami output.

6.4 Three findings

Finding 1: Joint bidirectional training is the single largest lever

Going from single-direction LoRA (C1, chrF++ 12.71) to joint bidirectional LoRA (C3, chrF++ 17.79) adds **+5.1 chrF points**. Critically, the joint variant also more than **doubles the FST acceptance rate from 23.25% to 45.45%** — the model is producing dramatically more morphologically valid Sami forms when also trained on the comprehension direction.

Mechanism hypothesis. With only 247 examples, single-direction LoRA on the harder direction (nob→sme) overfits to surface patterns and degrades general translation capability — C1's chrF++ (12.71) is *worse* than the untrained zero-shot baseline (C0, 15.65). Joint training appears to act as a regularizer: the comprehension direction (sme→nob) requires the model to align Sami input with Norwegian semantics, and that alignment signal counteracts the surface overfitting in the generation direction. This is consistent with the broader literature on multi-task regularization in low-resource NMT [30, 31, 32] but is, to our knowledge, the first quantitative demonstration on Sami specifically.

Finding 2: Soft FST-constrained decoding is a no-op under greedy decoding

C0 vs C0+FST and C1 vs C2 produce **bit-exact identical predictions and metrics**. The FST processor as currently implemented applies a 0.5-logit downward bias to subsequent tokens after a word-completing token has been accepted as morphologically invalid. Under deterministic greedy decoding (`do_sample=False, num_beams=1`), this 0.5-logit penalty is too weak to flip the top-1 token in any of the 47 test sentences for either model. The penalty fires (we logged the signal); it just does not reach the threshold where it would change the argmax.

This is consistent with the hard-vs-soft distinction in the constrained-decoding literature [23, 24, 26].

This is a methodological finding with a clean mechanism: soft logits processing is incompatible with greedy decoding for morphological constraints. Three corrections are available — (a) hard masking of FST-rejected continuations rather than soft penalty, (b) sampling-based decoding so the modified distribution actually changes outputs, (c) beam search with FST validation as part of the beam scoring function. We discuss (a) in §8.6 future work.

Finding 3: Joint training does affect FST decoding (C3)

C3 (joint + FST) is the *only* configuration in which FST decoding interacts with a model whose output distribution is already in a regime where the soft penalty matters. We cannot fully separate the joint-training effect from the FST-decoding effect in this single configuration; the proper ablation (joint-LoRA without FST) is enumerated in §8.6 future work.

6.5 Length and FST distribution analysis

Beyond aggregate FST acceptance, the per-sentence distribution is informative. Table 12 shows the FST-acceptance histogram (10% bins) across the 47 test sentences for each Track-C variant.

Variant	0–10%	10–20%	20–30%	30–40%	40–50%	50–60%	60–70%	70–80%	80–90%	90–100%
C0 zero-shot	14	16	11	3	1	2	0	0	0	0
C1 LoRA single	18	6	7	9	1	3	3	0	0	0
C3 joint + FST	**7**	**0**	**5**	**6**	**6**	**9**	**11**	**3**	**0**	**0**

Table 12. Per-sentence FST acceptance histogram (number of sentences in each 10% bin, n=47).

C0's distribution peaks at 10–20% with only 6 sentences exceeding 30% acceptance. C1 is bimodal — 18 sentences at 0–10% (degraded) and 9 at 30–40% (modestly improved) — illustrating the mixed effect of single-direction overfitting documented in §6.4. **C3's distribution is qualitatively different: it peaks at 60–70% (11 sentences), with an additional 9 sentences at 50–60%. The number of sentences exceeding 50% acceptance jumps from 2 (C0) and 6 (C1) to 23 (C3) — a half-of-test-set median above the soft validity threshold.**

Per-source breakdown (chrF++) further verifies the gain is broad-based and not driven by a single source genre. The full breakdown is in `results/per_source_breakdown.json`. Both children's-book sources (Du bestemmer and Masse) and the Paper:story literary excerpts show C3 dominance over C1 and C0.

6.6 Qualitative example for Track C

Variant	Output
Source (nob)	*Samtykke.*
Reference (sme)	*Miehtan.*
C0 zero-shot	Samtykkwa.
C1 LoRA single	Sáhttuvá.
C3 LoRA joint+FST	Muiddus.

Table 13. Track C qualitative example (id=gisting-eval/generated/dubestemmer:36). Note the visible difference: C0 produces noisy mixed-language output, C1 over-commits to surface Sami patterns without semantic alignment, C3 produces noticeably more coherent Sami including correctly-inflected forms validated by the FST.

7. Track D — Pedagogical demo: lær-bort-samisk

We implemented three small scenarios to demonstrate the *teach-don't-speak* deployment frame, structurally analogous to Te Hiku Media's Papa Reo design [22]. The full demo output is in `paper/appendix_demo_tutor.md` and Appendix B.

Architectural choice. All Sami wordforms shown are generated **directly by Apertium-sme's finite-state transducer** [9, 10], not by the LLM. The LLM's role in a real product would be to produce the surrounding pedagogical explanations *in the learner's L1 (Norwegian)* — generating exercise prompts, explaining grammatical concepts, scaffolding learning. The Sami output itself is morphologically guaranteed because it is FST-derived from the curated Giellatekno-Divvun lexicon.

This sidesteps the principal concern raised by Sami language activists about generative AI for endangered and Indigenous languages [33, 34]: the AI does not "speak" Sami. It uses a curated linguistic resource to show correct Sami, framed by L1 explanations.

- **Scenario 1: Verb conjugation drill.** Given lemma `boahtit` ("to come"), generate the full present-tense paradigm across nine person/number combinations (Sg/Du/Pl × 1/2/3). All nine forms — 1Sg `boađán`, 2Sg `boađát`, 3Sg `boahtá`, … 3Pl `bohtet` — generated by uralicNLP in <100 ms.
- **Scenario 2: Noun declension table.** Given lemma `gáhkku` ("cake/bread"), generate the seven Sami case forms in singular: Nom `gáhkku`, Gen `gáhku`, Acc `gáhku`, Ill `gáhkkui`, Loc `gáhkus`, Com `gáhkuin`. Essive returned no form — an honest signal that the FST does not have full coverage for that case for this lemma.
- **Scenario 3: Word-by-word translation help.** Given the Norwegian sentence "Jeg kommer hjem.", produce a morphological alignment: jeg → mun (`mun+Pron+Pers+Sg1+Nom`), kommer → boađán (`boahtit+V+IV+Ind+Prs+Sg1`), hjem → ruoktut (`ruoktut+Adv`).

8. Discussion

8.1 The Sami-stakeholder question

This work was deliberately conducted in Phase 1 *without* institutional collaboration with Sámediggi or other Sami language institutions. This was a strategic choice rather than an oversight: the goal of Phase 1 was to establish a reproducible technical baseline using only openly licensed tooling and data, before approaching institutional partners with concrete results in hand rather than hypotheses.

The cost of that choice is real. Without native-speaker validation, we cannot make any claim about whether our generated Sami is *natural* — only that substantial fractions of it are *morphologically valid* per Apertium-sme. The gap between FST-acceptance and naturalness is unbounded; a sentence may be morphologically perfect and semantically nonsensical or culturally tone-deaf. Phase 2 work, with appropriate institutional collaboration, must close this gap.

8.2 Implications for Norwegian AI policy

Two policy-relevant claims follow from our findings.

(a) FLORES-200 is not a defensible evaluation standard for Sami compliance. Procurement processes that rely on "FLORES-200 coverage" as a proxy for multilingual capability are, by construction, not measuring what they claim to measure with respect to Sameloven §3 obligations. Norwegian public-sector AI procurement should require explicit measurement on a Sami-inclusive evaluation set, even a small one — and the reproducible 47-pair evaluation we release here can serve as a starting baseline pending the development of larger gold-standard sets in Phase 2.

(b) Tokenizer choice has direct fiscal consequences. The 4–5× tokenization penalty under Qwen-stock is a direct cost multiplier on every Sami token processed. For administrative-text deployment specifically — which is noun-heavy and therefore subject to the most aggressive penalty (§4.4) — replacing the tokenizer alone, even without retraining, is an actionable cost reduction with no accuracy regression.

8.3 Implications for low-resource NMT methodology

Three methodological observations have implications beyond the Sami case study.

- **Joint bidirectional training as regularizer.** Our Track C result — that single-direction LoRA on 247 pairs degrades quality below zero-shot baseline, while joint bidirectional LoRA on the same pairs improves it substantially — should generalize to other genuinely low-resource pairs where the model has weak prior on at least one direction. Practitioners should default to joint training over single-direction at this scale.
- **Soft constrained decoding requires non-greedy search.** Our Finding 2 is independent of Sami specifics; it falls out of the math of greedy argmax under bounded logit perturbation. Any work using soft-penalty constrained decoding with greedy

generation should expect the same null result, and should switch to hard masking, sampling, or beam search.

- **FST-as-oracle is cheap.** The 10,000-validations-per-second throughput we observe with cached uralicNLP makes FST validation a free engineering primitive at decoder throughputs. The bottleneck for FST-aware methods is therefore design, not compute.

8.4 Limitations

- Test set is 47 pairs from one source family (children's books and short literary excerpts in Apertium's corpus). Generalization to news, official, or spoken-language domains is unmeasured.
- The Apertium-sme-nob parallel corpus we use is mixed-provenance: most test sentences come from the gisting-eval/ directory, which is intended for MT-system evaluation and may include curated MT output as part of the alignment. We treat the corpus as human-aligned but acknowledge that it is not fully gold human-translated.
- No native-speaker quality validation; the FST acceptance rate is an upper bound on naturalness, not a measurement of it.
- We evaluate only North Sami. Lule, South, and the smaller Sami languages have less FST infrastructure and would require separate work.
- The FST-constrained logits processor uses word-boundary validation, not token-boundary. A morphologically invalid wordform still appears in output before the FST signal fires; the signal merely discourages continuing in that direction.
- No round-trip consistency loss tested. A natural extension would be true cross-lingual KL regularization [38] under sme-paraphrase versus nob-paraphrase of the same semantic content.
- Compute budget capped at one 16 GB consumer GPU; scale-up effects beyond 3B parameters untested.
- Single seed for all translation experiments. Multi-seed validation (n≥3) is standard practice [37] but was outside the wall-clock budget for this Phase 1 release.

8.5 Ethics statement

This work raises questions specific to Indigenous-language technology that we want to address explicitly.

Consent and stewardship. All training data used in this work is openly licensed (CC-BY-SA-4.0 Wikipedia, GPL-3 Apertium parallel, CC-BY-2-FR Tatoeba). However, openly licensed is not the same as community-stewarded in the sense of [22, 33]. The Sami community has not consented to this specific use of openly licensed Sami text for LLM fine-tuning, because we did not ask. This is the principal ethical limitation of Phase 1, and is the explicit motivation for Phase 2's institutional engagement.

Risk of low-quality generated Sami. Generative AI for Indigenous languages poses a documented risk of polluting the linguistic landscape with low-quality output [33, 34]. We have designed Track D specifically to sidestep this risk: the demo's Sami output is FST-generated, not LLM-generated. We do not advocate deploying a Track-C-style end-to-end LLM translator

into production without Phase 2 native-speaker validation; the Track C results are research baselines, not deployment recommendations.

Power asymmetry. tenki is a Norwegian-majority technology venture. Building Sami-language tech as a Norwegian-majority venture, even with the best intent, replicates the power asymmetry that has characterized Norwegian–Sami relations historically. Phase 2's commitment to institutional collaboration with Sámediggi and Giellatekno-Divvun is intended in part to redistribute that asymmetry; we acknowledge that this commitment is partial and will require sustained effort to honour.

8.6 Future work

Four follow-up directions are directly motivated by the empirical findings of this paper.

(F1) A2′ — proper morphological pre-segmentation. §4.3 identifies that naive FST-token seeding fails because special tokens reserve vocabulary slots without affecting BPE merge order. The corrected design uses the FST to insert explicit boundary markers (e.g., _) between stem and inflectional suffix in the training corpus, then trains BPE on the segmented text. We expect this to recover most of the +93% Unigram advantage (A3) while retaining BPE's deterministic encoding properties.

(F2) Hard-masking FST decoder. §6.4 Finding 2 establishes that soft logits processing is incompatible with greedy decoding. The corrected approach is *hard masking*: at every word-boundary token, preemptively zero the logit of any token that would extend the partial wordform into a state not reachable by any valid Apertium-sme analysis. This requires building a trie of valid wordform prefixes from the FST and indexing it by tokenizer output [23, 24] — engineering work, not research, but necessary for the FST signal to actually fire under greedy decoding.

(F3) The joint-without-FST ablation. §6.4 Finding 3 acknowledges that we cannot separate the joint-training effect from FST decoding in C3. Running "C3 minus FST" (joint-bidirectional LoRA, no FST decoding) would isolate the contribution of joint training alone. We expect the chrF++ gain to remain (joint training is the dominant factor) but the FST acceptance rate to return toward C1 levels (≈23%), confirming that FST decoding contributes the morphological-validity portion of the C3 result.

(F4) Phase 2 with Sami-stakeholder engagement. All quantitative gains above are upper-bounded by the absence of native-speaker quality validation (§8.4). Phase 2 would pursue collaboration with Sámediggi, Sámi Giellagáldu, and the Giellatekno-Divvun research groups at UiT to (a) validate generated output against fluent-speaker judgments, (b) extend coverage to Lule and South Sami where FST infrastructure exists, and (c) develop a governance protocol for community-curated training data following the Te Hiku Media / Papa Reo model [22].

9. Conclusion

We presented a four-track empirical study of LLM techniques for North Sami on a single consumer GPU. The headline results: domain-trained 32k tokenizers crush the Qwen-stock 151k tokenizer by 73–94% on Sami compression, with the largest gains on the morphologically richest classes (nouns -63%, verbs -62%); joint bidirectional LoRA more than doubles the FST acceptance rate of generated Sami (17.97% → 45.45%) compared to the zero-shot baseline; and a deliberately small whitepaper-scale ablation produces three negative results with clean mechanisms — naive FST-token seeding fails, soft FST decoding is a no-op under greedy decoding, and single-direction LoRA on the harder direction can degrade quality below zero-shot.

Two strategic claims for Norwegian AI policy follow. First, FLORES-200 [1] is **not a defensible evaluation standard** for any deployment that must comply with Sameloven §1-5 — it excludes the language. Second, the deployment frame most consistent with Sami language activists' concerns and with the Te Hiku Media tradition [22] is *teach-don't-speak*: the LLM scaffolds learning in Norwegian while delegating every Sami output to the FST stack curated by Giellatekno-Divvun [7, 9, 10, 11]. We demonstrate this design in Track D.

Phase 2, with institutional Sami-language collaboration, will be necessary to convert these technical baselines into deployment-grade systems. The technical baseline itself, however, is reproducible from raw downloads in under one hour by anyone with a 16 GB consumer GPU. We hope this lowers the barrier to subsequent work on Sami AI substantially enough that the next paper can come from a much wider community than the one Phase 1 was conducted in.

Acknowledgements

The Apertium-sme [10] and Giellatekno-Divvun [7, 11] finite-state morphology and lexical resources, accessible via uralicNLP [9], are the load-bearing linguistic infrastructure for this work. Decades of open-licensed scholarship by Trond Trosterud, Sjur Moshagen, Lene Antonsen, Linda Wiechetek, Mika Hämäläinen, and many others made every FST-validated generation in this paper possible. The Te Hiku Media team and the broader Indigenous-tech community established the deployment-design principles we attempt to honour in Track D.

The technical infrastructure underlying our experiments — HuggingFace transformers [21], peft [16], bitsandbytes [15], sacrebleu [37], sentencepiece [5], tokenizers [3], pyhfst — is the product of open-source work spanning many institutions and contributors. All experiments were run on a single consumer NVIDIA RTX 5070 Ti (16 GB).

References

[1] Costa-jussà, M. R., Cross, J., et al. (NLLB Team). (2022). No Language Left Behind: Scaling Human-Centered Machine Translation. arXiv:2207.04672.

[2] OpenLanguageData. (2024). FLORES+ multilingual evaluation benchmark. https://openlanguagedata.org/.

[3] Sennrich, R., Haddow, B., & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. ACL.

[4] Kudo, T. (2018). Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. ACL.

[5] Kudo, T., & Richardson, J. (2018). SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing. EMNLP demo.

[6] Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H., & Mikolov, T. (2016). FastText.zip: Compressing text classification models. arXiv:1612.03651.

[7] Trosterud, T. (2006). Grammatically based language technology for minority languages. In Saxena, A., & Borin, L. (eds.), Lesser-Known Languages of South Asia. Mouton de Gruyter.

[8] Wiechetek, L., Hämäläinen, M., & Antonsen, L. (2019). Apertium-sme-nob: Rule-Based Machine Translation for North Sami to Norwegian Bokmål. Proceedings of the Workshop on the Use of Computational Methods in the Study of Endangered Languages.

[9] Hämäläinen, M. (2019). UralicNLP: An NLP Library for Uralic Languages. Journal of Open Source Software, 4(37), 1345.

[10] Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., & Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. Machine Translation, 25(2), 127–144.

[11] Lindén, K., Silfverberg, M., Axelson, E., Hardwick, S., & Pirinen, T. A. (2011). HFST — Framework for Compiling and Applying Morphologies. Systems and Frameworks for Computational Morphology, Springer LNCS.

[12] Lov om Sametinget og andre samiske rettsforhold (Sameloven). (1987, with subsequent amendments). LOV-1987-06-12-56. Norwegian Parliament.

[13] Norwegian Constitution. (Kongeriket Norges Grunnlov). §108 (Sami clause, originally §110a, renumbered 2014).

[14] Lov om språk (språklova). (2022). LOV-2021-05-21-42. Norwegian Parliament. (Mållov / Norwegian Language Act, regulating use of Bokmål and Nynorsk in public administration.)

[15] Dettmers, T., Lewis, M., Belkada, Y., & Zettlemoyer, L. (2022). LLM.int8(): 8-bit Matrix Multiplication for Transformers at Scale. NeurIPS.

[16] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-Rank Adaptation of Large Language Models. ICLR.

[17] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient Finetuning of Quantized LLMs. NeurIPS.

[18] Popović, M. (2017). chrF++: words helping character n-grams. Proceedings of the Second Conference on Machine Translation (WMT).

[19] Rust, P., Pfeiffer, J., Vulić, I., Ruder, S., & Gurevych, I. (2021). How Good is Your Tokenizer? On the Monolingual Performance of Multilingual Language Models. ACL.

[20] Petrov, A., La Malfa, E., Torr, P. H. S., & Bibi, A. (2024). Language Model Tokenizers Introduce Unfairness Between Languages. NeurIPS.

[21] Yang, A., Yang, B., Zhang, B., et al. (2025). Qwen 2.5 Technical Report. arXiv:2412.15115.

[22] Mahelona, K., Leoni, G., Duncan, S., & Thompson, M. (2023). OpenAI's Whisper is another case study in Colonisation. Te Hiku Media. https://blog.papareo.nz/whisper-is-another-case-study-in-colonisation/. See also: Te Hiku Media. Kaitiakitanga License (Papa Reo data license).

[23] Willard, B. T., & Louf, R. (2023). Efficient Guided Generation for Large Language Models. arXiv:2307.09702. (outlines)

[24] Beurer-Kellner, L., Fischer, M., & Vechev, M. (2024). Guiding LLMs The Right Way: Fast, Non-Invasive Constrained Generation. ICML.

[25] Karlsson, F., Voutilainen, A., Heikkilä, J., & Anttila, A. (1995). Constraint Grammar: A Language-Independent System for Parsing Unrestricted Text. Mouton de Gruyter.

[26] Poesia, G., Polozov, A., Le, V., Tiwari, A., Soares, G., Meek, C., & Gulwani, S. (2022). Synchromesh: Reliable Code Generation from Pre-trained Language Models. ICLR.

[27] Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. LREC.

[28] HuggingFace. (2024–2026). Datasets library deprecation notice for repository scripts. https://huggingface.co/docs/datasets/.

[29] Goyal, N., Gao, C., Chaudhary, V., Chen, P.-J., Wenzek, G., Ju, D., Krishnan, S., Ranzato, M., Guzmán, F., & Fan, A. (2022). The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation. TACL.

[30] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., et al. (2017). Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. TACL.

[31] Aharoni, R., Johnson, M., & Firat, O. (2019). Massively Multilingual Neural Machine Translation. NAACL-HLT.

[32] Sennrich, R., & Zhang, B. (2019). Revisiting Low-Resource Neural Machine Translation: A Case Study. ACL.

[33] Lewis, J. E., Abdilla, A., Arista, N., Baker, K., Benesiinaabandan, S., Brown, M., Cheung, M., et al. (2020). Indigenous Protocol and Artificial Intelligence Position Paper. The Initiative for Indigenous Futures and CIFAR. Honolulu.

[34] Bird, S. (2020). Decolonising Speech and Language Technology. COLING.

[35] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. ACL.

[36] Post, M. (2018). A Call for Clarity in Reporting BLEU Scores. WMT.

[37] Post, M., et al. (2018–2024). SacreBLEU. https://github.com/mjpost/sacrebleu.

[38] Bornea, M., Pan, L., Rosenthal, S., Florian, R., & Sil, A. (2021). Multilingual Transfer Learning for QA Using Translation as Data Augmentation. ACL.

Appendix A: Reproduction protocol

```
git clone <repo>

cd sami-llm-research

make reproduce
```

Every result file in `results/*.json` records: timestamp, full hyperparameter set, SHA-256 of every input file, Python+torch+CUDA versions, GPU device name, and library versions. Datasets are pinned by direct upstream URL. Random seeds are set on `random`, NumPy, PyTorch CPU+CUDA at process start; CUDA determinism is enabled where supported.

A.1 Compute summary

Step	Wall-clock
Data acquisition (Wikipedia, Apertium clone, Tatoeba dumps)	~3 min
Track A: train 4 tokenizers (3.83 MB sme corpus)	1 min
Track A: eval 4 tokenizers	1 min
Track B + C: 8 evaluations + 3 LoRA training runs	55 min
Track D: FST tutor demo (boahtit, gáhkku, translation)	<1 min
Figure generation (4 publication PNGs)	<5 sec
Extra analyses (qualitative, per-source, per-PoS, FST distribution)	<2 min
Total wall-clock (RTX 5070 Ti, 16 GB)	**≈ 1 hour**

Table A1. Compute summary for full reproduction.

Appendix B: lær-bort-samisk demo output

B.1 Verb conjugation: boahtit ("to come"), present indicative

Person	Norwegian pronoun	Sami form
1Sg	jeg	**boađán**
2Sg	du	**boađát**
3Sg	han/hun	**boahtá**
1Du	vi to	**bohte**
2Du	dere to	**boahtibeahtti**
3Du	de to	**boahtiba**
1Pl	vi	**boahtit**
2Pl	dere	**boahtibehtet**

Person	Norwegian pronoun	Sami form
3Pl	de	**bohtet**

B.2 Noun declension: gáhkku ("cake/bread"), singular

Case	Tag	Form
Nominativ	Nom	**gáhkku**
Genitiv	Gen	**gáhku**
Akkusativ	Acc	**gáhku**
Illativ (til)	Ill	**gáhkkui**
Lokativ (i/på/fra)	Loc	**gáhkus**
Komitativ (med)	Com	**gáhkuin**
Essiv	Ess	*[no form]*

B.3 Translation help: "Jeg kommer hjem."

Norwegian	Sami lemma	Tag	Generated form
Jeg	mun	Pron+Pers+Sg1+Nom	**mun**
kommer	boahtit	V+IV+Ind+Prs+Sg1	**boađán**
hjem	ruoktut	Adv	**ruoktut**

Appendix C: Prompt templates

All translation prompts use Qwen 2.5's standard chat template via the tokenizer's apply_chat_template helper. The system and user messages follow this pattern:

C.1 sme → nob

```
system: "Du er en oversetter. Oversett gitt nordsamisk tekst til norsk bokmål. Svar bare med oversettelsen."

user:   "Oversett til norsk bokmål:\n<sme source>"
```

C.2 nob → sme

```
system: "Du er en oversetter. Oversett gitt norsk bokmål-tekst til nordsamisk. Svar bare med oversettelsen."
```

```
user:   "Oversett til nordsamisk:\n<nob source>"
```

C.3 Five-shot variant (B1)

Same system message; user and assistant turns alternate with 5 fixed example pairs taken from the first 5 examples of the training set, followed by the actual test input as a final user turn. The example seed (5 examples, fixed seed=42) is held constant across all 47 test sentences for B1.

Appendix D: FST-constrained logits processor

Pseudocode for the decoder. Full implementation in scripts/fst_decode.py.

```
class FSTConstrainedLogitsProcessor:     def __init__(self, tokenizer,
language="sme", penalty=5.0, validation_window=1):         self.tok = tokenizer
self.lang = language         self.penalty = penalty         self.window =
validation_window         self._last_check = {}  # per-beam memoization      def
__call__(self, input_ids, scores):         if not URALIC_AVAILABLE:
return scores         for b in range(input_ids.shape[0]):             tail_ids =
input_ids[b, -32:].tolist()             text = self.tok.decode(tail_ids,
skip_special_tokens=True)             words = WORD_RE.findall(text)             if
len(words) < self.window + 1:                 continue             last_complete =
words[-2]  # -1 is in-progress             beam_key = id(input_ids) ^ (b << 16)
if self._last_check.get(beam_key) == last_complete:                 continue
self._last_check[beam_key] = last_complete             if not
_is_in_fst(last_complete, self.lang):                 scores[b] = scores[b] - 0.1 *
self.penalty  # soft bias         return scores  @lru_cache(maxsize=200_000) def
_is_in_fst(word, language):     if not URALIC_AVAILABLE: return True       # graceful
fallback     if not word or len(word) < 2: return True   # short tokens passthrough
try:         analyses = uralicApi.analyze(word.lower(), language)         return
len(analyses) > 0     except Exception:         return True   # fail-open on FST
error
```

Note that this implementation applies a soft penalty of 0.5 logit (0.1 × penalty=5.0). As documented in §6.4 Finding 2, this is too weak under greedy decoding. The corrected hard-masking variant — preemptively zero the logit of any token that would extend the partial wordform into an FST-unreachable state — is enumerated in §8.6 (F2).

Appendix E: Additional sample predictions

We include two additional Track-C sample sentences (beyond the §6.6 example) to allow readers to inspect output style across variants directly.

Variant	Output
Source (nob)	*Utdanning*

Variant	Output
Reference (sme)	*Oahppu*
C0 zero-shot	Utdanninga
C1 LoRA single	Ottáđa
C3 LoRA joint+FST	Eakkáduagasuohta

Table E1. Track C sample (id=gisting-eval/generated/masse:8, source: Masse (children's book MT-eval)).

Variant	Output
Source (nob)	*Signing*
Reference (sme)	*Sivdnádallan*
C0 zero-shot	Signe
C1 LoRA single	Sáhkuváđđa
C3 LoRA joint+FST	Riikkašuvvo

Table E2. Track C sample (id=gisting-eval/generated/masse:25, source: Masse (children's book MT-eval)).

All predictions for all variants on all 47 test sentences are released as `results/<variant>_predictions.jsonl` with one record per sentence: id, source, reference, prediction. Readers can inspect any subset directly.