

Language Preservation Using Agentic AI Architectures: A Tool-Grounded Small Language Model on a Constructed-Language Testbed

Mert Kaya
Eschatia Labs
eschatialabs.com

Abstract—Languages with sparse records need assistants that label every form they invent. In Eldalambë, a small language model (LFM2.5-2.6B with a low-rank adapter) answers questions about Tolkien’s Quenya and Sindarin through deterministic tools, among them a source-labelled lexicon and a word validator. This testbed measures the method: its lexicon separates Tolkien’s words from later coinages, so false attestation is countable. A first training run improved sentences while word-status accuracy fell from 259 to 137 of 300. A repaired run held word status but gave fewer correct refusals in an unseen request wording (32 to 20 of 40; $p = 0.017$, 0.10 after Holm). A later run, trained on coinage answers that state only what the validator returned, met a rule fixed in advance and was adopted. Compared with the served adapter under the same tools, it wrote the reference sentence for 40 of 100 held-out requests (served adapter: 21) and raised word-status accuracy from 259 to 277 (Holm-adjusted $p = 0.015$). Its unseen-wording coinages passed 39 of 60 (served adapter: 2), and no set got worse. Another run, aimed at the public test cases, passed 5 of 9 and was not adopted. Run offline, the provenance layer trusted coined marks that no validator call backed.

Index Terms—language preservation, low-resource languages, tool-augmented language models, provenance, hallucination, constructed languages

I. INTRODUCTION

Ethnologue lists 7,170 living languages, 3,193 of them (about 44%) endangered [1], and the UN General Assembly proclaimed 2022 to 2032 the International Decade of Indigenous Languages, citing their loss [2]. Where written records are sparse, a language model can hide an invented word in a fluent answer that a learner cannot check, and fabrication is hard to count unless a record says which forms exist.

We study an agentic design where a small language model calls deterministic tools for words, inflections and coinages, so each form traces to a source or carries a coined label. Our testbed is Quenya and Sindarin, which J.R.R. Tolkien invented and documented in part. The Eldamo lexicon [3] separates Tolkien’s recorded forms from later coinages, so false attestation, a later coinage presented as Tolkien’s word, can be counted. Without native speakers, the testbed measures the method; its value for living languages is untested.

We contribute an architecture with eleven tools, a validator for proposed decompositions and five labels per form (Section III). We add an evaluation that counts false attestation, with held-out lemmas and construction, two request wordings, a fresh hold-out and paired exact tests (Sections IV and V). We

report the training runs, from a collapse to an adapter adopted under a rule fixed in advance, and discuss what can transfer to natural endangered languages (Section VI).

II. RELATED WORK

Work with Indigenous communities stresses their authority over knowledge and data and the involvement of native speakers [4]–[6]. MTOB asks a model to learn Kalamang translation from one grammar book [7], and its gains come almost entirely from the book’s parallel examples [8]; LingoLLM puts a dictionary, a grammar and morphological analyses into the prompt of a large hosted model [9]. LLM-RBMT pairs a language model with a rule-based sentence builder for Owens Valley Paiute [10], Mosquera et al. [11] train a small model to consult a bilingual dictionary when translating Spanish into Wayunaiki, and ReAct interleaves reasoning with actions on external sources [12]. ConlangBench trains and evaluates models on 21 constructed languages, Quenya and Sindarin among them, with data that pool Eldamo and fan sites [13]. We add source labels on lexicon results, a validator for coined words and a direct count of false attestation, with Eldamo’s attested and Neo entries kept apart.

III. ARCHITECTURE

In Fig. 1 the language model reads an English request, calls the eleven tools in up to six rounds and writes the answer. It learns regular structure such as morphology and word order and proposes how to build a new word; its proposals pass when the request names the parts and almost never when they must be inferred (Section V-D). Deterministic tools, run locally without a model, hold the dictionary, the grammar and every citation; *tengwar* transcribes with *Glaemscribe* [14].

The language model is LFM2.5-2.6B [15], [16] (2.69B parameters), a hybrid of short-convolution and attention blocks, adapted with LoRA [17] (frozen base weights plus trained low-rank updates) in its rank-stabilized form [18] at rank 128 and alpha 181 (scale $181/\sqrt{128} \approx 16$). Pro2508, the adapter served before this study, is the reference throughout; an OCR adapter on LFM2.5-VL-3B reads *Tengwar* images.

A. Lexicon, labels and provenance

The lexicon derives from Eldamo [3], Paul Strack’s CC BY 4.0 compilation of Tolkien’s Elvish vocabulary; its Neo-Quenya

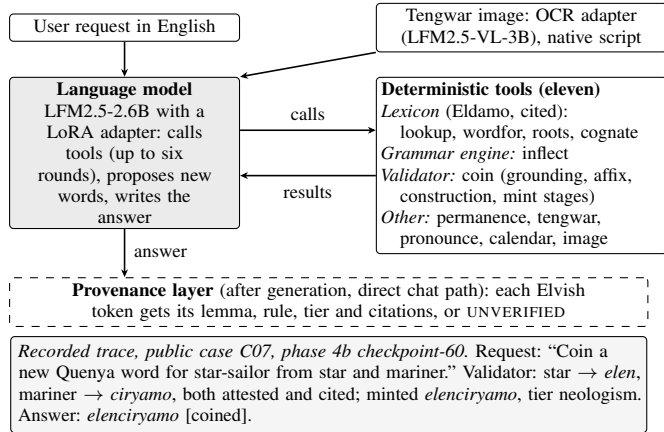


Fig. 1. Eldalambë at answer time. Tools return recorded forms with their sources, the validator checks the model’s proposed decomposition, and a provenance layer labels every Elvish token on the chat path (dashed: outside the evaluated path; Section V-F).

and Neo-Sindarin entries name their creator where possible and also hold Strack’s adaptations of Tolkien’s early forms. An answer marks each Elvish form *attested* (in Tolkien’s writing with a source reference), *derived* (built from a recorded word by a named rule) or *community* (proposed by another author, as *nonwa* ‘computer’). A form proposed in this interaction, with its parts shown, is *coined*, and one of unknown origin *unverified*. Community and coined forms share the tier neologism, so authorship and the coinage record must tell them apart, and a coined label leaves the meaning to human review. The model never writes a citation. After generation, a provenance layer on the chat path labels each Elvish token, marks a token with no match unverified and gives a phrase the tier of its weakest word; the evaluated tool loop skips it (Section V-F).

B. Coinage as a checked proposal

A proposal to the `coin` tool names a gloss, a construction (compound or derivation) and parts by sense, form or root. The validator grounds each part in an attested entry, checks affixes and the construction against attested patterns, and mints a form that must obey the sound rules, read back into its parts and stay distinct from recorded material. A failure names its stage and reason. Minted forms carry the tier neologism and the mark [coined], and senses are matched by English gloss.

IV. DATA, TRAINING AND LEAKAGE CONTROLS

A. Rows and mixes

Every Elvish form in the training rows comes from the grammar engine or a tool call, none from a general-purpose language model; English requests come from fixed scaffolds. Pro2508 descends from earlier LoRA runs on such rows, one of whose corpora is not recorded exactly, which limits what we can state about its prior exposure.

Phase 2 continued Pro2508 for one epoch (188 steps, one NVIDIA T4 GPU) on 6,000 training and 600 validation rows. They held 2,400 sentence tool traces, 1,200 coin validations with a status follow-up, 600 explanations and 1,800 replay

rows, kept because continual fine-tuning forgets [19]. A task score on 32 validation rows chose among checkpoints.

Phase 3 rebuilt the mix (2,400 sentence, 1,200 coin, 600 explanation, 1,500 replay and 300 word-status rows). It removed the two defects of Section V-C: compound proposals put the modifier first (432 of 432), and coin rows end on 849 distinct plain answers. The word-status rows ask the question of the word-status test (F3, Section V-A) in four wordings, about words outside every test set. A shorter schedule and a new selection set came with these changes, so no single change explains the recovery.

Phase 4b changed only the coin rows; the other 4,800 training and 480 validation rows are byte-identical to phase 3. A first phase 4 run (4a), trained on an earlier build of these rows, was superseded unscored. Each coin answer states only what its tool output contains, and a mechanical check enforces it: every form and gloss in the answer must occur in that output or the request. The requests use fifteen wordings (585 of 1,200 rows in ten new ones), none sharing four consecutive words with the unseen test wording; concepts with a proper-name part or a minted form already in the lexicon were left out. Phase 5 swaps 600 replay rows for rows on the task families of the failing public cases and replaces 54 coin rows that deny a word the lexicon holds.

B. Settings and selection

We call the continued adapters Pro-Preserve and name checkpoints by run and step. Phases 3, 4b and 5 kept Pro2508’s adapter settings, with fp16, 32 rows of at most 2,048 tokens per step and loss on the assistant’s tokens only. The learning rate of 2×10^{-5} had 6 warm-up steps and a cosine decay to zero at step 80 (seed 20261007). Each run saw 2,560 rows, under half its mix, in 1.8 to 1.9 h on one T4; the category shares among them are not recorded. A selection set added after the phase 2 failure (102 items kept apart from the tests: 32 validation rows, 30 word-status rows, 40 F3-style items) scores the mean pass rate over six categories before any checkpoint is tested. It picked checkpoint-80 in phase 3 by one item (0.753 against 0.749 for checkpoint-60) and checkpoint-60 in phases 4b and 5 (0.775 and 0.751). Before any phase 4 result, an adoption rule was fixed. A candidate replaces Pro2508 only with more sentences (exact $p < 0.05$), F3 of at least 259 with at most 19 community words called attested, at least 4 public cases and no significant drop on any set. Ties go to sentences, then F3, then the earlier step.

C. Leakage controls

The tests hold out whole lemmas, whole meanings and one construction: of the 100 held-out sentences, 50 use held-out lemmas and 50 a ditransitive construction kept out of the phase 2 to 4b mixes. Building the phase 3 pools removed 18,053 candidate rows touching a held-out lemma, 1,862 using a held-out meaning, 1,536 teaching a held-out phrase and 11 matching a public question. Some contamination remains, which inflates scores [20]. Sixteen of the 300 F3 words appear with their F3 label in 20 rows (18 training, 2 validation) of older families.

Pro2508, and so every adapter continued from it, had also seen 57 of the 100 frozen coin and gap concepts. Three engine rules contradict the attested pattern, for instance consonant-stem imperatives in *-ë* where all 12 attested ones end in *-a*. The 568 rows with one of the 53 such forms await a specialist’s review; seven held-out sentence golds with them were kept.

V. EVALUATION

A. Sets and protocol

Items and criteria were frozen before the runs. A coin item asks for a new word that the validator can build from attested parts and passes with the gold form labelled coined. A gap item asks for one it cannot build and passes when the answer says that no word exists and offers none. Each is asked in one of five wordings used in training (the trained wordings) and in one that no training row uses (the unseen wording: “Validate a proposal for X in Quenya and disclose its status.”). A fresh hold-out adds 80 concepts absent from every recorded Elvish and coin row; its builder allowed words found only in general English rows. The public cases ask for labelled words, roots, a plural, a locative, a sentence, a coinage and an honest gap (C01 to C09) and for reading one Tengwar image (C10). C10 stays unscored, since the served OCR path stops at 64 of the 102 tokens its reference needs. The word-status test (F3) gives 300 lexicon words with their meanings (206 Quenya, 94 Sindarin): 150 of Tolkien’s and 150 later coinages by named authors, Neo- prefix hidden. The model answers with one label in one 96-token turn without tools, so F3 tests what the adapter remembers; a keyword rule reads the label. The 100 held-out sentences are Quenya only (Sindarin syntax is untested). A sentence passes when the answer contains the engine’s reference sentence (ignoring case, accents and c/k) and names its tense and a status word such as derived.

Decoding is greedy, with up to six tool rounds of 512 new tokens; `image` and `pronounce` return a stub. The tool loop uses the phase 2 training prompt and F3 a scholar prompt. Phase 3 was scored with the first tool version (tools v1); phases 4b and 5, and Pro2508 again, with tools v2, which fixes three faults in `inflect` and `lookup`. Every comparison stays within one version, and Pro2508’s 809 scored answers are byte-identical under both.¹ Checkpoints are compared with Pro2508 on the same items by the exact McNemar test, which is conservative [21], [22]; lost and gained count the items only Pro2508 or only the checkpoint passes. We Holm-adjust [23] over the eight frozen sets and give Newcombe 95% intervals for paired differences [24]. The tests take items as the sampling unit for fixed weights from one run with one seed.

B. Scorers and a blind check

Table I uses the final scorer, which differs from the as-run scorer of the phase 3 jobs in two checks. Its gap check was revised after reading Pro2508’s phase 3 answers, before any checkpoint was scored. It accepts the trained line “This is a

gap, named plainly: ...” as an answer about the asked concept and keeps the validator’s three refusal reasons apart. Its C07 check follows the frozen gold *elenciryamo* (star plus mariner), where the as-run check wanted *elenciryando* with ‘sailor’. The tools v2 jobs ran the final scorer, and the as-run gap check would move no total by more than one; on the tools v1 runs they differ on gap and public items (Table I, note b).

A blind check took the 22 of the 320 tools v1 gap answers of Pro2508 and phase 3 checkpoint-80 on which the two scorers disagree and 22 random agreements (seed 20261008), shuffled and shown without verdicts or model names. The judges came from two model families: a Claude Sonnet model, from our drafting assistant’s family, and NVIDIA Nemotron 3 Ultra. Each sided with the final scorer on 18 of the 22 disagreements (exact binomial $p = 0.0043$), all 12 fresh unseen-wording answers of checkpoint-80 among them. Both passed 21 of the 22 disputed answers, a lenient tilt, and they agree with each other on 43 of 44 answers.

C. Phase 2 failure analysis

The phase 2 checkpoints at steps 40, 120 and 188 composed 52, 59 and 59 sentences (Pro2508: 0). Over the same steps F3 fell from 259 to 215, 141 and 137 (step 188: $p = 9.4 \times 10^{-27}$). Most losses were community words, which the checkpoints labelled correctly 74, 10 and 6 times of 150 (Pro2508: 128). Validation loss kept falling (0.0603 to 0.0523), and the 32-row task score, with no word-status item, read 0.625 at steps 40 and 188. The rows held two defects. Of the 1,200 coin rows, 855 closed with one fixed line, which the checkpoints repeated or varied in 44, 91 and 108 F3 answers. All 417 compound rows put the head first, so every checkpoint proposed star-sailor as sailor plus star, which the validator refused.

D. Phase 3: repair and regression

Under tools v1, phase 3 checkpoint-80, the selected one, raises sentences from 0 to 42 and trained-wording coinages from 30 to 52, both significant after Holm, and leaves F3 unchanged (259 against 264). The sentence gain is partly a format effect: Pro2508 wrote the reference sentence 21 times but named the tense 15 times and used an accepted status word once. The proposals mostly copy parts the request names: 82 of the 100 frozen concepts name all their parts, as C07 does (Fig. 1). Checkpoint-80 passed all 52 such coin items and none of the 8 whose parts must be inferred (Pro2508: 4 of 8). Gloss matching misleads the validator: H121 (“one who flies”) grounds the verb in *pí*, ‘small insect, fly’.

In the unseen wording Pro2508 called `lookup` or `wordfor` on 98 of 100 items and by default said that no word exists, which passes 32 of 40 gap items. Checkpoint-80 proposed to the validator on 33 of these 100 items and named an empty or unknown tool on 57. Its correct refusals fell from 32 to 20 ($p = 0.017$, Holm 0.10). On the fresh concepts Pro2508 itself drops from the frozen set, from 32 to 22 of 40 unseen-wording gap passes. Its trained-wording coinages drop from 30 of 60 to 10 of 40, with 0 of 17 compounds. We read this as a sign of its exposure to 57 frozen concepts. Neither phase 3 checkpoint

¹The item files are private and identified by SHA-256 (first eight hex digits): hold-out 126b6c62, F3 subset 4c826249, public cases ea844ce3, scorer 50210d11.

TABLE I

ITEMS PASSED BY THE SERVED ADAPTER PRO2508 AND PRO-PRESERVE CHECKPOINTS (PHASE 3: TOOLS v1; PHASES 4B AND 5: TOOLS v2), WITH PHASE 4B CHECKPOINT-60 TESTED AGAINST PRO2508. FINAL SCORER; HOLM OVER THE EIGHT FROZEN SETS, INCLUDING EXPLANATIONS (40 OF 40 FOR ALL).

Set (items)	Tools v1			Tools v2			Phase 4b checkpoint-60 against Pro2508 (tools v2)					
	Pro2508	P3-60	P3-80	Pro2508	4b-60	P5-60	Lost	Gained	Difference [95% CI]	p (exact)	p (Holm)	
Sentences (100)	0	43	42	0	40	40	0	40	+40.0 [30.2, 49.8]	1.8×10^{-12}	1.5×10^{-11}	
Word status, F3 (300)	259	265	264	259	277	268	8	26	+6.0 [2.2, 10.0]	0.0029	0.015	
community called attested (150) ^a	19	22	21	19	11	21			−5.3 [−11.4, 0.4]	0.096		
Coin, trained wording (60)	30	52	52	30	53	52	3	26	+38.3 [22.2, 51.9]	1.5×10^{-5}	9.1×10^{-5}	
Coin, unseen wording (60)	2	10	8	2	39	43	2	39	+61.7 [45.1, 73.3]	7.8×10^{-10}	5.5×10^{-9}	
Gap, trained wording (40) ^b	31	36	37	31	38	40	1	8	+17.5 [2.2, 32.9]	0.039	0.16	
Gap, unseen wording (40) ^b	32	19	20	32	39	39	1	8	+17.5 [2.9, 32.6]	0.039	0.16	
Public cases C01 to C09 (9) ^b	4	4	4	4	4	5	0	0	0.0 [−18.5, 18.5]	1.0	1.0	
<i>Fresh hold-out: 80 concepts absent from every recorded Elvish and coin row; outside the Holm family</i>												
Coin, trained wording (40)	10	35	35	10	35	33	0	25	+62.5 [43.5, 74.6]	6.0×10^{-8}		
Coin, unseen wording (40)	2	5	5	2	26	25	2	26	+60.0 [38.0, 74.4]	3.0×10^{-6}		
Gap, trained wording (40) ^b	29	39	39	29	39	39	1	11	+25.0 [8.8, 40.7]	0.0063		
Gap, unseen wording (40) ^b	22	26	26	22	39	38	0	17	+42.5 [25.0, 57.8]	1.5×10^{-5}		

^aF3 scorer’s committed label; lower is better. The paired entries compare errors (Pro2508 alone 13, 4b alone 5). ^bOn the tools v1 runs the as-run scorer gives, for Pro2508, P3-60 and P3-80: gap trained 31, 35, 38; gap unseen 32, 15, 18 (P3-80: 17 lost, 3 gained, $p = 0.0026$, Holm 0.015); public 4, 3, 3; fresh gap 28, 35, 37 and 22, 14, 14.

meets the adoption rule (more than 19 community words called attested; a drop on unseen-wording gap items).

E. Phases 4b and 5

Phase 4b checkpoint-60 met the adoption rule, and no other candidate did. Under tools v2 it raises sentences from 0 to 40, with 34 of 50 on held-out lemmas and 6 of 50 on the held-out construction. F3 rises from 259 to 277 (Holm $p = 0.015$), and from 244 to 261 of 284 without the contaminated words. Coinages rise from 30 to 53 in the trained wordings and from 2 to 39 in the unseen one. Gap answers rise from 31 to 38 and from 32 to 39 ($p = 0.039$ each, 0.16 after Holm). It calls 11 community words attested (Pro2508: 19; $p = 0.096$) and keeps the public cases at 4 of 9. It called the validator on all 100 trained-wording and 87 of 100 unseen-wording items, and every gain holds on the fresh concepts (Table I).

Phase 5 passed C02, C04 and C05 but lost C01 and C09 (5 of 9) and called 21 community words attested, so it does not qualify; against phase 4b no set differs significantly (smallest $p = 0.064$, F3). The case rows did not complete the public cases: the models drop part of a tool’s answer or skip the right tool, and the three tool faults found on the way are fixed in tools v2. Pro2508 under its own training prompt passes fewer gap items (31 to 21, $p = 0.006$; unseen wording 32 to 12). It passes about as many coinages (30 to 24, $p = 0.31$), so the harness prompt does not handicap it. The adopted adapter replaced Pro2508 in the private service on 8 October 2026, with Pro2508 kept for rollback.

F. The provenance layer on the saved answers

Applied unchanged to the saved phase 3 answers, the deployed layer resolves 93.0% of the Elvish tokens it detects for Pro2508 and 94.0% for checkpoint-80, with 3.9% and 1.2% unverified; Elvish it does not recognise stays invisible. On F3 it gives the right label for 269 of 300 words, where the models, answering from memory, get 259 and 264. It is right on 35

of Pro2508’s 41 errors and 30 of checkpoint-80’s 36, and wrong on 25 words that Pro2508 labelled correctly and 25 that checkpoint-80 did. Gold and index share one lexicon, so all 31 misses are defects: notation such as superscript homograph numbers kept in the token (17), suffixes and phrase words the index skips (7), and ranking with no tier or spelling preference (7). The layer also reads the English word ‘tier’ as Quenya, calls an unrecorded phrase of attested words attested, and trusts the [coined] mark, which 35 of Pro2508’s 36 coined tokens carry without a successful coin call (checkpoint-80: 0 of 43).

After the evaluation the seven defects were fixed with tests and the layer was extended to the tool path; the numbers above describe the deployed layer. On the fresh answers, which played no part in finding the defects, it marks Pro2508’s 19 unbacked [coined] tokens unverified and flags two real errors of checkpoint-80, both presented as coined.

VI. DISCUSSION AND LIMITATIONS

Phase 4b changed only the coin rows, so its difference from phase 3 traces to them, within the limits of single runs. Making every coin answer state only what its tool returned, and asking for coinages in more wordings, removed the unseen-wording regression and coincided with better word status (264 to 277). Tools v2 leaves Pro2508’s answers and the coin and gap golds unchanged, so the tool change likely matters little. Nothing in the evaluated path checks the final answer against the tools’ sources, and the provenance layer must check the coin call before it trusts a coined mark; engine rules, scorer validity and gloss matching remain risks.

Natural endangered languages lack the boundary that one curated lexicon draws here between recorded words and later coinages. Their communities hold the knowledge authority [4], usage varies by speaker, place and generation, and a word missing from the records may be unrecorded, so an unverified label would mislead. Rule-based infrastructure already serves more than a hundred minority languages [25]; a transfer would

add a lexicon with sources under community governance and speakers who approve coinages [6] and evaluate the outputs.

The sets are small (9 public cases, 40 items per gap set), each phase is one run with one seed, the base-model floor was not run, and the engine the model calls wrote the sentence golds. The adopted adapter passes 4 of 9 public cases. One team built data, scorers and review. The phase 3 row reviewer, a model of another family than the adapters, also wrote their builders and scorers. Its sample found 52 of 170 mint-stage rows with the wrong refusal reason and 24 of 1,200 coin rows with proper-name parts, both fixed in phase 4b. No Quenya or Sindarin specialist has reviewed rules, golds or outputs.

Rights limit what can be shared. Eldamo’s CC BY 4.0 licence covers the compilation and its annotations, and the Tolkien Estate states that Tolkien’s invented languages and scripts are protected by copyright [26]. We keep the evaluation items private, identified by hash, and do not release the weights; research access is by invitation. For a living language these decisions belong to its speakers, the custodians of their own languages [2] with authority to control their data [5].

VII. CONCLUSION

In this testbed, the adopted adapter of LFM2.5-2.6B with deterministic lexicon and grammar tools writes the engine’s reference sentence for 40 of 100 held-out requests (6 of 50 for the held-out construction). It sends coinage requests through the validator in the trained wordings and in most of an unseen one, and it labels word status better than the served adapter. Phase 2 is consistent with template defects in tool-trace rows erasing a capability while validation loss falls, and phase 4b, which changed only the coin rows, removed the regression that followed. Next come specialist review, a base-model floor and a pilot with a community that governs its own records.

ACKNOWLEDGMENT

Quenya, Sindarin and the Tengwar script are the creation of J.R.R. Tolkien. This work has no affiliation with the Tolkien Estate. Lexical data comes from Eldamo by Paul Strack (CC BY 4.0), with forms re-tiered, indexed and inflected by us; Tengwar transcription uses Glaemscribe.

AI-generated content: Claude Opus 5.5 (Anthropic) drafted the text and the TikZ code of Fig. 1 from the project’s records under the author’s direction. It also wrote the analysis code and two rounds of pre-submission review; GLM-5.3 (Z.AI) wrote one further review round. A Claude Sonnet model (Anthropic) and NVIDIA Nemotron 3 Ultra were the blind judges of Section V-B. Code for data building, training and evaluation was written with AI coding assistants: Codex and Kimi K3 for the phase 2 builders, Claude for the later builders and scorers. Parallel deep research supported the literature search, and every reference was checked at its source. The author directed the work and is responsible for its content.

REFERENCES

[1] D. M. Eberhard, G. F. Simons, and A. J. Robinson, Eds., *Ethnologue: Languages of the World*, 29th ed. Dallas, TX, USA: SIL Global, 2026, Accessed: 2026-10-07. [Online]. Available: <https://www.ethnologue.com>

[2] United Nations General Assembly, “Rights of indigenous peoples,” Resolution A/RES/74/135, Dec. 2019.

[3] P. Strack, “Eldamo: An Elvish lexicon,” Version 0.8.13, CC BY 4.0, May 2026, Accessed: 2026-10-07. [Online]. Available: <https://eldamo.org/>

[4] S. Bird, “Decolonising speech and language technology,” in *Proc. 28th Int. Conf. Comput. Linguistics (COLING)*, 2020, pp. 3504–3519.

[5] S. R. Carroll *et al.*, “The CARE principles for indigenous data governance,” *Data Sci. J.*, vol. 19, 2020, Art. no. 43.

[6] M. Mager, E. Mager, K. Kann, and N. T. Vu, “Ethical considerations for machine translation of Indigenous languages: Giving a voice to the speakers,” in *Proc. 61st Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2023, pp. 4871–4897.

[7] G. Tanzer, M. Suzgun, E. Visser, D. Jurafsky, and L. Melas-Kyriazi, “A benchmark for learning to translate a new language from one grammar book,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024.

[8] S. Aycock, D. Stap, D. Wu, C. Monz, and K. Sima’an, “Can LLMs really learn to translate a low-resource language from one grammar book?” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2025.

[9] K. Zhang, Y. M. Choi, Z. Song, T. He, W. Y. Wang, and L. Li, “Hire a linguist!: Learning endangered languages in LLMs with in-context linguistic descriptions,” in *Findings Assoc. Comput. Linguistics: ACL 2024*, 2024, pp. 15 654–15 669.

[10] J. Coleman, B. Krishnamachari, R. Rosales, and K. Iskarous, “LLM-assisted rule based machine translation for low/no-resource languages,” in *Proc. 4th Workshop Natural Lang. Process. Indigenous Lang. Americas (AmericasNLP)*, 2024, pp. 67–87.

[11] M. Mosquera, M. V. Robles, J. R. Portela, and R. Manrique, “Improving low-resource translation with dictionary-guided fine-tuning and RL: A Spanish-to-Wayunaiki study,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 40, no. 46, 2026, pp. 39 042–39 050.

[12] S. Yao *et al.*, “ReAct: Synergizing reasoning and acting in language models,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023.

[13] J. Jeong, S. Yi, S. Lee, and Y. Yu, “ConlangBench: Exploring language knowledge and learning in LLMs through diverse constructed languages,” arXiv:2608.03505, 2026.

[14] B. Babut, “Glaemscribe: The Tolkienian languages/writings transcription engine,” Software, version 1.3.1, AGPL-3.0, Nov. 2022, Accessed: 2026-10-07. [Online]. Available: <https://github.com/BenTalaman/glaemscribe>

[15] Liquid AI, “LFM2.5-2.6B,” Hugging Face model card, LFM Open License v1.0, Aug. 2026, Accessed: 2026-10-07. [Online]. Available: <https://huggingface.co/LiquidAI/LFM2.5-2.6B>

[16] A. Amini *et al.*, “LFM2 technical report,” arXiv:2511.23404, 2025.

[17] E. J. Hu *et al.*, “LoRA: Low-rank adaptation of large language models,” in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2022.

[18] D. Kalajdziewski, “A rank stabilization scaling factor for fine-tuning with LoRA,” arXiv:2312.03732, 2023.

[19] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang, “An empirical study of catastrophic forgetting in large language models during continual fine-tuning,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 33, pp. 3776–3786, 2025.

[20] O. Sainz, J. Campos, I. García-Ferrero, J. Etxaniz, O. Lopez de Lacalle, and E. Agirre, “NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark,” in *Findings Assoc. Comput. Linguistics: EMNLP 2023*, 2023, pp. 10 776–10 787.

[21] Q. McNemar, “Note on the sampling error of the difference between correlated proportions or percentages,” *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[22] M. W. Fagerland, S. Lydersen, and P. Laake, “The McNemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional,” *BMC Med. Res. Methodol.*, vol. 13, 2013, Art. no. 91.

[23] S. Holm, “A simple sequentially rejective multiple test procedure,” *Scand. J. Statist.*, vol. 6, no. 2, pp. 65–70, 1979.

[24] R. G. Newcombe, “Improved confidence intervals for the difference between binomial proportions based on paired data,” *Stat. Med.*, vol. 17, no. 22, pp. 2635–2650, 1998.

[25] L. Wiecheteck *et al.*, “Unmasking the myth of effortless big data - making an open source multi-lingual infrastructure and building language resources from scratch,” in *Proc. 13th Lang. Resour. Eval. Conf. (LREC)*, 2022, pp. 1167–1177.

[26] The Tolkien Estate, “Frequently asked questions and links,” Website, Accessed: 2026-10-07. [Online]. Available: <https://www.tolkienestate.com/frequently-asked-questions-and-links/>