ConlangCrafter: Constructing Languages with a Multi-Hop LLM Pipeline

Morris Alper[*1,2] **Moran Yanuka**[*1] **Raja Giryes**[1] **Gašper Beguš**[3]

*Equal contribution [1]Tel Aviv University [2]Carnegie Mellon University [3]UC Berkeley

https://conlangcrafter.github.io

Figure 1: **ConlangCrafter outputs,** showing random examples of sample sentences and glosses in diverse conlangs generated with our multi-hop LLM pipeline. Each language is designed to be internally consistent, while being typologically unique in its phonology and morpho-syntax.

Abstract

Constructed languages (*conlangs*) such as Esperanto and Quenya have played diverse roles in art, philosophy, and international communication. Meanwhile, foundation models have revolutionized creative generation in text, images, and beyond. In this work, we leverage modern LLMs as computational creativity aids for end-to-end conlang creation. We introduce *ConlangCrafter*, a multi-hop pipeline that decomposes language design into modular stages – phonology, morphology, syntax, lexicon generation, and translation. At each stage, our method leverages LLMs' metalinguistic reasoning capabilities, injecting randomness to encourage diversity and leveraging self-refinement feedback to encourage consistency in the emerging language description. We construct a novel, scalable evaluation framework for this task, evaluating metrics measuring consistency and typological diversity. Automatic and manual evaluations demonstrate ConlangCrafter's ability to produce coherent and varied conlangs without human linguistic expertise.

1 Introduction

"pʰán dzáwali-li a-ga-galúnta-mi áta-li."

every language-INTR EVID.NEUT-IPFV-be_a_world-3SG.INTR he/she/it-INTR.

"Every language is a world." (conlang)

As humans have long imagined alternative methods of communication, the art of constructing languages has evolved into a creative and scholarly pursuit (Schreyer, 2021). Constructed languages, or *conlangs*, span the gamut from artistic endeavors to bring fictional worlds to life (e.g. J.R.R. Tolkien's Elvish and Dothraki in Game of Thrones) and attempts to bridge international divides for worldwide communication (e.g. Esperanto) to tests of philosophical ideas (e.g. Lojban and Toki Pona). Conlangers may spend years or even decades designing their creations, marvels of linguistic ingenuity requiring Sisyphean effort to achieve the scope and complexity of natural languages.

As foundation models are now being used for various creative tasks, including generation of novel artistic content (Chakrabarty et al., 2024; Teleki et al., 2025), we ask: Can these models be used to create conlangs? By introducing the new paradigm of *computational conlanging*, we investigate three core research questions: **(RQ1)** Can LLMs generate internally consistent linguistic systems that are distinct from those seen during training? **(RQ2)** Can LLMs generate diverse yet coherent outputs for this creative application? **(RQ3)** Can we develop objective, scalable evaluation metrics for this task with no ground-truth?

To tackle these questions, we propose a multi-hop reasoning-based LLM pipeline, *ConlangCrafter*. This constructs a language layer by layer, using insights from linguistic typology and documentation (Genetti, 2018; wal, 2013), resulting in languages with diverse phonologies and grammars as shown in Figure 1. Our checklist-based prompting method with injected randomness ensures diverse, typologically interesting output

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 9318–9349
July 2-7, 2026

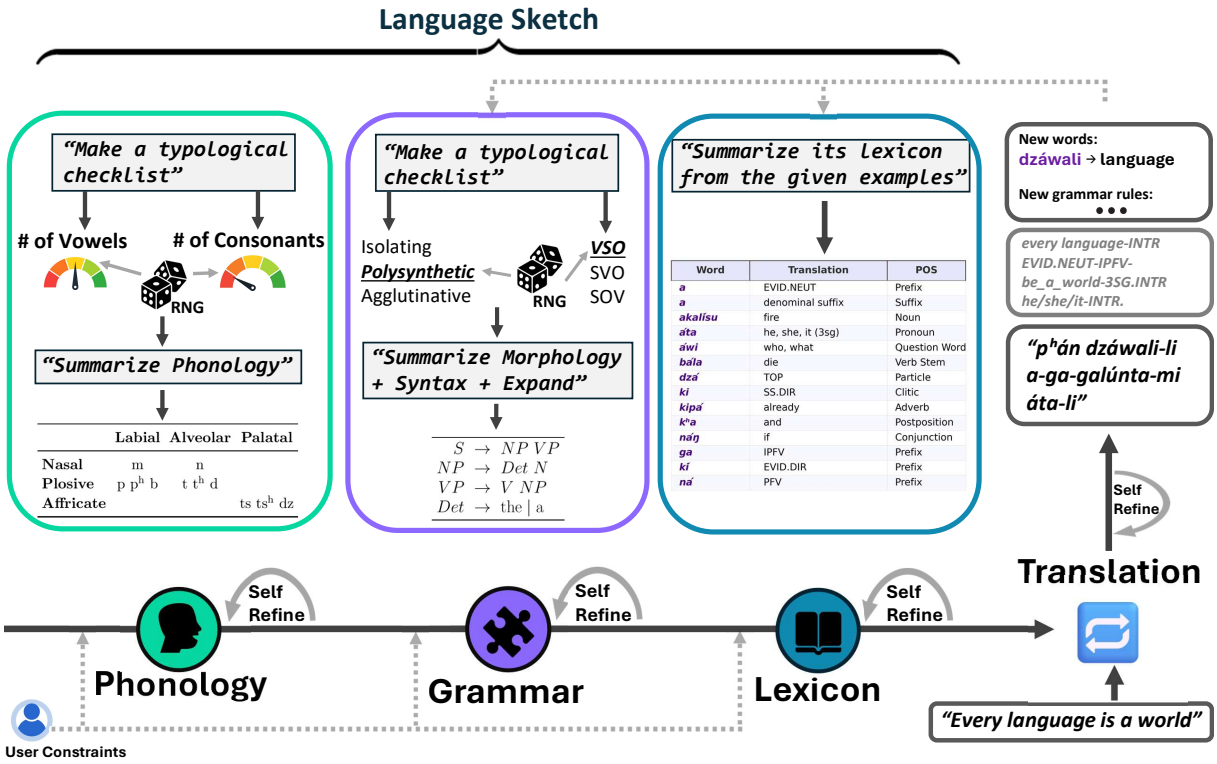

Figure 2: **Our Method.** ConlangCrafter constructs languages with a multi-hop LLM pipeline, generating a language sketch as a sequence of linguistic layers (phonology, grammar, lexicon). We encourage typological diversity using checklist-based prompting and a random number generator (RNG), and enhance internal consistency via self-refinement. ConlangCrafter conditions on this sketch to translate and gloss new sentences, potentially including new lexical items and grammar points which may be dynamically added back to the language sketch. Prompts above are abridged and some intermediate steps are omitted.

languages, while our proposed *self-refine* loop enhances the consistency of generated language descriptions. We also propose a novel framework for *constructive translation* into conlangs where lexemes and grammar may not be fully specified a priori. Finally, we introduce a novel evaluation framework that systematically assesses the quality of generated conlangs and their translations with scalable, automatic metrics. To support the validity of automatic evaluation, we assess agreement with manual expert judgements, as well as performing qualitative evaluation.

Computational conlanging offers significant theoretical and practical value. As a challenging logically-grounded task, it tests the complex meta-linguistic and logical reasoning abilities of LLMs (Begus et al., 2025) on an inherently out-of-distribution language; because the target language does not yet exist, the model cannot rely on memorized training data. From a practical perspective, it may serve as a computational creativity aid for hobbyist or professional conlangers, promising to ease laborious aspects of conlanging such as lexicon generation while allowing for creative control. Moreover, it has direct applications in procedural world or society generation in systems such as open-world video games. Finally, we see potential applications of our methodology to low-resource languages, where our framework's focus on logical consistency with set grammatical rules may be relevant for languages documented mostly through written grammars.

2 Method

ConlangCrafter is an LLM-driven stochastic pipeline which generates a random conlang either fully from scratch, or in accordance with optional user-input specifications. It leverages LLMs' ability to perform creative generation that is not factually grounded, exploiting "hallucination" as a desirable feature, as well as exploiting their meta-linguistic knowledge (Begus et al., 2025) (i.e. ability to explicitly reason about language). As these models struggle out-of-the-box to produce diverse and coherent outputs, ConlangCrafter is designed to encourage typological diversity and to mitigate

contradictions while splitting the overall task of conlang generation into tractable sub-problems. Our full method is illustrated in Figure 2, and we proceed to describe its overall structure (Section 2.1) and core concepts (Sections 2.2 and 2.3).

2.1 Overall Pipeline

ConlangCrafter has the following core components

- An LLM M. In practice this is a large reasoning model, i.e. a modern LLM using chain-of-thought inference-time scaling (Guo et al., 2025).
- A memory bank S, referred to as the *language sketch*. This consists of free text containing the language's structure description, which may be retrieved from or dynamically altered throughout the pipeline. Examples of languages sketches are provided in Appendix B.1.
- An optional user-input string c, with specifications or constraints on the generated conlang. By default, this is the empty string $\emptyset$.

The pipeline proceeds in two stages:

Stage A: Language Sketch Bootstrapping. We first generate an initial description of the language's core structure and store it in S. Generating this description is a challenging task since it must be both sufficiently detailed and as internally consistent as possible, and our results show that a single LLM prompt is insufficient to adequately generate S. To this end, we address the problem with a multi-hop pipeline that incrementally updates S with linguistic information.

Linguistic theory identifies aspects of language such as phonology as objects of independent study (Genetti, 2018), and language documentation materials such as Visser (2022) commonly split a language's description into sections based on these aspects. Following this practice, we divide a language into three key layers: *phonology*, *grammar* (morpho-syntax), and *lexicon*. Because these layers depend on one another, we generate them sequentially (e.g., phonology precedes grammar to provide word forms). This structure parallels multi-hop approaches to other complex reasoning tasks with LLMs (Khot et al., 2022). We also note that this minimalistic structure deliberately omits additional potential layers describing linguistic aspects such as semantics, pragmatics, and orthography, which could be addressed in future work.

Each layer is produced through multiple substeps, prompting M with the current state of S, optional user input c, and the target language aspect; the result is incorporated back into S. To ensure diversity and consistency, we apply randomness injection (Section 2.2) and self-refinement (Section 2.3). Once S is initialized, it can guide translation and be expanded dynamically.

Prompts at each stage include a slot for optional user input, specifying that it takes priority if present. This permits user control for applications such as generating full conlangs from high-level ideas or initial hand-crafted conlang sketches[1].

Stage B: Constructive Translation. Given S, ConlangCrafter translates and glosses new texts by conditioning on the explicit language description, building a corpus while dynamically updating S as needed. We term this task *constructive translation* as it has the unique aspect that the existing language description may be under-specified and require new, creative additions to translate a given text. This is unlike low-resource translation (Tanzer et al., 2023; Zhang et al., 2024; Zhang et al.) and glossing (Ginn et al., 2024a,b), in which explicit hallucination is undesirable.

During translation, ConlangCrafter prompts the model M with a source text t and instructions tasking it translate it consistently with the language description in S. Its output includes fields for the translation and interlinear gloss, as well as optional fields for new lexical items and grammar rules, which may be output as needed to resolve under-specification in S. The latter can be added back to S to expand the language description and ensure that they are used consistently in future translations (although we without this feedback loop; see Section 3.2.2). Logical consistency of these translations with S is enhanced using self-refinement (Section 2.3). This iterative process builds a translation corpus while expanding the language's grammar and lexicon.

2.2 Randomness Injection.

While LLMs display awareness of language typology, they often fail to produce diverse outputs across runs (Hopkins and Renda, 2023), resulting in limited typological variety in language sketches S. To address this, we inject randomness into the phonology and grammar stages during sketch bootstrapping. At the start of each stage, the LLM M generates a typological checklist of ten linguistic

[1]Online repositories such as `https://conlang.fandom.com/wiki/Portal:Main` contain many partial conlang sketches, as developing complete grammars and lexicons requires substantial time and effort.

features, each with five multiple-choice options. A random number generator (RNG) then selects one option per feature, and M instantiates the language description accordingly. This leverages the model's metalinguistic knowledge of typology while offloading control of diversity to the external RNG.

2.3 Self-Refinement.

Internal consistency is crucial, as contradictions saved to the language sketch may propagate to later stages, and translations must adhere to the language's constructed grammar. Therefore, we handle violations of logical consistency by leveraging a key observation – evaluating generated content is often more straightforward than producing it, enabling iterative refinement through self-feedback mechanisms where LLMs model critique and revise their own outputs (Wu et al., 2024; Simonds et al., 2025; Madaan et al., 2023). We adopt a similar paradigm: a critic model identifies errors and ambiguities in generated content, and an editor model revises accordingly. These are both implemented with the base LLM M, prompted with S and the text under revision (as well as the list of identified errors in the case of the editor model). This process is repeated iteratively until no further issues are detected or a maximum number of iterations is reached to prevent infinite revision cycles.

3 Experiments

We test the utility of ConlangCrafter and its potential for creative, consistent conlang generation. We present our experimental setup (Section 3.1); quantitative evaluation framework (Section 3.2); results (Section 3.3); ablations (Section 3.4); and examples, analysis, and applications (Section 3.5).

3.1 Experimental Setup

For base LLMs, we use DeepSeek-R1 (Guo et al., 2025), and Gemini 2.5 (Comanici et al., 2025) in two variants (Flash, Pro). For automatic evaluations, we use OpenAI o3[2] as the judge LLM, selected to avoid introducing bias by both generating and evaluating with the same model. For metric calculations in our experiments, we sample $\sim$20 languages with 10 test sentences each (see Section 3.2.2); we confirm the sufficiency of this sample size with a statistical significance and effect size analysis (Section 3.3). Lacking a prior method for full conlang generation, we create a reasonable baseline via a single-stage generation method. This attempts to generate a full language sketch and translations of given sentences in our format with a single prompt, without multi-hop reasoning or iterative self-refinement. Further experimental details are provided in the appendix.

[2]Described in OpenAI's system announcement

3.2 Quantitative Evaluation Framework

A key challenge to evaluating conlang generation is the lack of an existing framework for this novel, partially subjective task. Unlike typical machine translation, there is no inherent ground-truth for languages which do not exist. Human evaluation for attributes such as logical consistency and diversity requires expert knowledge and painstaking attention to detail, failing to scale to a sufficient sample size of languages for rigorous evaluation.

To tackle this challenge, we develop a comprehensive automatic evaluation framework using the LLM-as-a-judge approach (Zheng et al., 2023; Gu et al., 2024) as a scalable proxy for expert evaluation. We support its overall validity by assessing agreement with manual expert judgments performed on a smaller scale, demonstrating moderate agreement with manual evaluation on this challenging task. Despite the known limitations of such automatic metrics, this framework addresses an existing gap by enabling large-scale, quantitative evaluation of computational conlanging methods, previously infeasible with manual evaluation alone.

3.2.1 Typological Diversity Analysis

To evaluate the breadth of typological variation captured by ConlangCrafter, we select a fixed set of $k = 16$ basic typological features from the World Atlas of Language Structures (WALS) (wal, 2013) covering fundamental aspects of language. These features are chosen to cover the most fundamental and broad range of basic typological dimensions differentiating natural languages. For example, the feature "Basic word order" (WALS 81) indicates the relative order of subject, verb, and object in basic sentences (e.g. SVO, SOV, VSO, etc.). The full list of features used is provided in the appendix. Each feature is assigned a categorical value (e.g., "SOV", "Postpositions", "Tonal"). While the majority of feature values can be determined, missing or underspecified values are recorded as empty for the purpose of diversity calculations.

We generate languages $L_1, L_2, \ldots, L_N$, use a judge LLM to encode each L_i as a one-hot vector $\mathbf{x}_i \in \mathbb{Z}_2^k$ over these features,

and calculate the **diversity score** $D_{\text{mean}} = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} \frac{\text{Ham}(\mathbf{x}_i, \mathbf{x}_j)}{k}$, where $\text{Ham}(\cdot, \cdot)$ is the Hamming distance. This reflects the average proportion of differing features between languages.

To ground these values in the distribution of existing languages, we also extract the same WALS features for all natural languages in the WALS database with sufficient coverage, yielding a sample of 1874 languages. We calculate the same diversity score on this sample and compare to values produced by ConlangCrafter and baseline methods.

3.2.2 Internal Consistency Evaluation

We evaluate internal consistency by testing for validity of a fixed set of translations conditioned on a language sketch. This tests whether the language's rules are logically valid and can be applied coherently. For each language, we prompt the model to translate a predefined set of $N_{t,t} = 10$ test sentences (via constructive translation, when evaluating ConlangCrafter). These are designed to cover a variety of syntactic structures (see Appendix A.4 for the full list of sentences). Each sentence is translated and evaluated independently, without re-adding new lexical items or grammar rules to the language sketch, to measure adherence to the original sketch. A separate judge-based evaluation prompt then assesses each of the generated translations to determine if its phonology, morphology, and syntax adhere to the rules specified in the language sketch. The **translation consistency rate** is defined as the ratio $\frac{N_{c,t}}{N_{t,t}}$ of correctly formed translations to total translations, averaged over all generated languages. While this is limited in equally penalizing minor and major errors, it nevertheless provides an indication of the degree to which the language can be parsed.

3.2.3 Manual Expert Evaluation

Human evaluation in our setting is only feasible on a small scale, as it requires cross-referencing with a language sketch in order to assess language traits and consistency of a translation with all relevant details. Unlike standard NLP annotation tasks, each judgment requires a holistic consistency check against a novel, custom-designed grammar, rather than comparison to a fixed reference. In addition, this may only be performed by experts with linguistic training to understand technical linguistic materials. To support the validity of our larger-scale automatic evaluation, we perform a manual human evaluation parallel to Sections 3.2.1 and 3.2.2 on all $N = 20$ languages and $N_{t,t} = 10$ translations output by a single model. We briefly describe the annotation procedure below, with further details and full task instructions provided in the appendix.

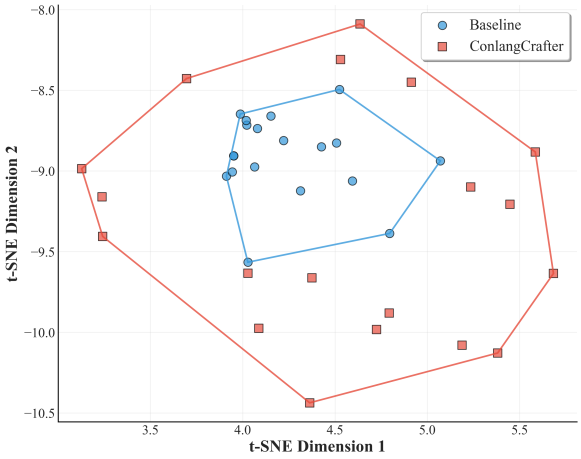

Figure 3: **t-SNE visualization of typological diversity.** Each point represents a generated language, where spatial proximity reflects typological similarity based on WALS features. The dispersed distribution of ConlangCrafter-generated languages highlights their higher typological diversity.

For evaluating diversity calculations, annotators received browsable HTML files containing each language's sketch and automatically inferred WALS features. They were instructed to label each inferred feature based on correctness given the language sketch. We report the error rate of automatic WALS features relative to these annotations as well as inter-annotator agreement (Cohen's κ).

Translation consistency was evaluated similarly, with browsable HTML files containing language sketches and candidate translations paired with interlinear glosses. An automated LLM judgement was displayed alongside each candidate as an auxiliary cue only; the annotation task instructions requested the annotators to rely on their own reading of the language sketch when scoring. While we note that this may potentially introduce a priming effect, our pilot tests found that performing this manual evaluation entirely from scratch was infeasibly labor-intensive. We adopt this design as a practical compromise, as well as noting that annotators readily corrected automatic judgements as reflected in agreement metrics. We report agreement (Spearman ρ) between mean manual and automatic scores. These are both on ordinal scales from 0=*fully inconsistent* to 2=*fully consistent* (noting that the final automatic metric in Section 3.2.2

	Gemini-2.5-Flash		Gemini-2.5-Pro		DeepSeek-R1	
Metric	**Baseline**	**Ours**	**Baseline**	**Ours**	**Baseline**	**Ours**
Diversity (↑)	0.25 ± 0.01	0.60 ± 0.01	0.26 ± 0.02	0.58 ± 0.01	0.35 ± 0.01	0.56 ± 0.02
Consistency (↑)	0.30 ± 0.04	0.38 ± 0.04	0.32 ± 0.05	0.54 ± 0.05	0.21 ± 0.04	0.39 ± 0.05

Table 1: **Typological diversity and consistency** comparison between the baseline method and ConlangCrafter, measured using the automatic metrics described in Section 3.2. Results are shown with ± stderr.

only counts fully consistent translations). We also report inter-annotator agreement (weighted κ).

Our manual evaluation was performed by two PhD students in linguistics with expertise in language documentation and typology, recruited through our institution. Participant compensation complied with applicable institutional policies. Each annotation required detailed expert analysis, involving cross-referencing between the generated sentence, its gloss, and the full language sketch. In total, the manual evaluation required approximately 35 hours of expert labor by PhD-level linguists. The goal of manual evaluation is therefore not to provide large-scale human scoring, but to validate that our automatic metrics meaningfully track expert judgments on this challenging task.

3.3 Results

Quantitative Results. Quantitative results are shown in Table 1, evaluating ConlangCrafter with base LLMs as well as the baseline method (Section 3.1). Our method outperforms the baseline in both diversity and consistency metrics. For diversity scores, all comparisons between the baseline and our method are highly significant (two-sided t-tests, $p < 10^{-6}$) with large effect sizes (Cohen's $d > 3.0$). While features may be underspecified, we note mean feature coverage of 94–98% across settings, indicating a negligible impact on diversity scores. For consistency scores, baseline comparisons are significant (two sided t-tests, paired by sentence) for Gemini-2.5-Pro ($p \approx 0.023$) and DeepSeek-R1 ($p \approx 0.014$) with large effect sizes ($d > 1.4$). For Gemini-2.5-Flash, the small effect ($d \approx 0.37$) does not reach significance ($p \approx 0.294$) at our sample size, but is consistent with findings that ConlangCrafter increases diversity without reducing consistency. In addition, we see moderate differences between the different LLMs in performance, with the stronger Gemini-2.5-Pro and DeepSeek-R1 performing better at this challenging task. Overall, generation of fully consistent translations remains a challenge, although our full method yields a better diversity-consistency tradeoff relative to the baseline.

In Figure 3, we apply t-distributed stochastic neighbor embedding (t-SNE) to visualize the pairwise Hamming distance matrix in two dimensions, where spatial proximity reflects typological similarity. The scattered distribution of languages across the 2D space visually confirms the high typological diversity compared with the baseline. These findings confirm that ConlangCrafter's randomness injection and modular prompting yield a diverse typological sample, even when evaluated against a fixed, expert-selected set of WALS features.

Comparison to Natural Languages. We also calculate diversity values for natural languages (Section 3.2.1), yielding $D = 0.43 \pm 0.003$, which falls between baseline and ConlangCrafter values in Table 1. All pairwise comparisons are statistically significant (t-tests, all $p < 10^{-6}$). This indicates that the baseline method generates languages with lower typological diversity than natural languages, while ConlangCrafter exceeds this level of diversity. Thus, despite our base LLMs exhibiting strong performance on general multilingual benchmarks, they fail to match the diversity of natural languages when used without our method, failing to leverage their knowledge of cross-lingual variation. By contrast, our method demonstrates successful exploration of the full typological space including novel feature combinations. For creative generation applications, exceeding natural language diversity is desirable, as it allows for imaginative world-building and testing of linguistic hypotheses with typologically unusual combinations.

To further assess novelty of generated conlangs relative to natural languages, we select the nearest neighbor to each generation in the WALS database of natural languages via Hamming distance. For ConlangCrafter with Gemini-2.5-Pro, these nearest neighbors have a mean of 56.4% matching comparable features (min=50.0%, max=68.8%), showing

Metric	Baseline	+MH	+MH+RNG	+MH+RNG+SR (Full)
Diversity (↑)	0.26 ± 0.02	0.52 ± 0.01	0.60 ± 0.02	0.58 ± 0.01
Consistency (↑)	0.32 ± 0.05	0.40 ± 0.05	0.43 ± 0.05	0.54 ± 0.05

Table 2: **Ablation study of ConlangCrafter's components** showing the cumulative effects of adding key components – Multi-Hop reasoning (MH), RNG randomness injection, and Self-Refinement (SR) – on automatic metrics over 20 languages generated by Gemini-2.5-Pro. Results are shown with ± stderr.

Temp.	RNG	Diversity	Consistency
0.6	✗	0.46 ± 0.02	0.59 ± 0.08
1.0	✗	0.50 ± 0.02	0.40 ± 0.09
1.4	✗	0.56 ± 0.02	0.34 ± 0.03
1.6	✗	0.54 ± 0.02	0.25 ± 0.06
0.6	✓	0.60 ± 0.02	0.43 ± 0.05

Table 3: **Comparison of diversity enhancement methods:** temperature sampling (above) and our RNG-based randomness injection (below). We achieve high diversity without the severe consistency penalty associated with high-temperature sampling. Self-refinement was disabled in this ablation.

that the conlangs exhibit highly novel typological feature combinations rather than replicating existing languages. For example, the four languages in Table 4 are closest to Yaqui (Uto-Aztecan, 52.9% match), Canela (Macro-Ge, 56.2%), Hausa (Chadic, 56.2%), and Axininca (Arawakan, 53.8%), respectively. These values reflect qualitative divergence in basic typological features; for example, the first language shares Yaqui's simple tone system, adjective-noun order, and question particle strategy, while differing in basic word order, adposition type, and consonant inventory size.

Manual Evaluation Results. For manual evaluation of diversity calculations, we examined automatic WALS features for all cases where annotators agreed (Cohen's $\kappa = 0.58$; moderate agreement). The most common issue was due to underspecification issues; 12% of potential features were mistakenly left unspecified, while 4% of features that were not specified had incorrectly inferred automatic values. Excluding underspecification issues, automatic feature extraction achieved 91% accuracy. This demonstrates that our diversity metric is based on largely accurate typological analysis, with the main existing limitation regarding the treatment of underspecified features.

For manual evaluation of translation consistency, we find significant agreement between manual and automatic metrics, with Spearman $\rho = 0.68$ ($p < 10^{-27}$). Inter-annotator agreement is quadratic weighted $\kappa = 0.43$ ($p < 10^{-4}$), indicating moderate agreement between annotators. This level of agreement is expected, as judgments require interpreting underspecified grammars and weighing minor versus major inconsistencies, decisions for which multiple valid analyses may exist. Overall, automatic scores tend to be stricter than manual evaluations, but still reasonably track human judgements when calibrated for this strictness level.

We contextualize both of these moderate agreement scores by noting that similar scores are standard in evaluations of natural language generation (Van der Lee et al., 2021), particularly for creative tasks which involve subjective reasoning and natural variability of human linguistic interpretation. In this context, exceptionally high agreement may be a negative indicator, signaling overly simplistic evaluation criteria (Amidei et al., 2018).

3.4 Ablation Analysis

In Table 2 we ablate key components of our system: multi-hop reasoning (MH), randomness injection (RNG), and iterative self-refinement (SR). We add each of these in turn to the baseline method (described in Section 3.1) and observe their cumulative effect. Multi-hop reasoning and randomness injection significantly improve diversity, key for generating typologically interesting and varied languages. This reduces consistency, as expected, since diverse outputs may be more logically challenging. However, self-refinement mitigates this issue, significantly increasing consistency approaching or beyond baseline levels. This matches qualitative observations that multi-hop reasoning and randomness injection are key for interesting, diverse outputs, while self-refinement succeeds in correcting many salient logical errors in generations.

To further test the importance of random-

Sample Sentences		Key Features
"The big dog is sleeping."	**"She will give him water."**	
tsɔ˩ gɔ-snat˥ kɻa˥˩ !a˥˩.θɔ˩ PRS 3SG.NHUM.S-sleep big dog.NOM	ʃi˩.t̪u˥-i˥˩ gɔ˩-|o/ do˩-sul˩-bo˥-dʒo˥ man-ACC 3SG.HUM.S-FUT 3SG.HUM.O-water-have-CAUS	Click consonants, polysynthetic morphology, noun incorporation, OVS word order, switch-reference, double marking
nɔ-k'a^{n}da jɔk'ɔ to-hoto ^{m}bulo hɛ PAT-sleep dog.SG.ABS 3SG.N.ABS-be.STAT big.ADJ PRF	se tʊ-nɔ-p'ajɛ hɛ wuwolo hɛ tu IRR AGT-PAT-give 3SG water 3SG to	ATR harmony, ejectives and prenasalized stops, active-stative alignment, VSO word order, ejective-conditioned word order changes
ˈg^{w}aŋk ˈk^{j}espə n^{j}o.skaˈm^{j}i ʃi big dog sleep 3S.PAT	n^{w}aˈco.l^{j}o ˈgr^{j}o.bə k^{w}o ʃi ʃi FUT give 3S.AGT 3S.PAT 3S.PAT	Frequent secondary articulations, isolating morphology, evidentiality marking, SOV word order with post-verbal particles marking person
ʂ̆ăŋ ʈ̪ə́m-jə́ ŋâ.ʈà-să-n dog big-ERG.ANIM sleep-IPFV.VIS.LOW-T-3SG	nɔ́.kà qá mă.n̪ă.a lə́m.ət kjâ.là.n FUT to man-ABS.ANIM water-ABS.INAN give-PFV.VIS-3SG	Vertical vowel system, dental-alveolar-retroflex distinction, split ergativity by aspect, tonal polarity in verb inflection, animacy-governed adposition order, serial verb constructions
User-controlled generation (user input in italics):		
"There are no consonant phonemes."	a-ɑ.o.-ɑ-úu̯-óu̯ 1.SUBJ-see-2.OBJ-PFV-DIRECT "I have seen you."	Non-phonemic glottal and pharyngeal onsets appear phonetically depending on syllable nucleus vowel quality.
"The language is produced by an alien cephalopod species. Phonemes are color values and gestures rather than consonants or vowels."	PB BR SRHC.PWHC(Peak) SRSW(Peak) PFV VIS CL1.AGT-move(V.PEAK) this.one(CL1) "Did this being move?"	"Chromemes" (color-based) have codes indicating dynamic features like static (S-) and pulsating (P-), and hue features like warm (-R) and cool (-B). "Kinemes" (gesture-based) such as HC involve curling tentacles. Contour "tones" indicate movement patterns through the water.

Table 4: **Qualitative examples** of ConlangCrafter-generated languages. The first four were generated unconditionally; the last two use user-input constraints (shown in italics), demonstrating user control and the ability to test creative ideas beyond attested natural languages.

ness injection, we compare to standard diversity-enhancing techniques reliant on sampling temperature, under our generation settings (without self-refinement). While increasing the sampling temperature (using nucleus sampling with $p = 0.95$) effectively increases variation, it operates at the token level rather than the typological level. As shown in Table 3, approaching the diversity score of our method (0.60) via standard sampling requires raising the temperature to $T \approx 1.4$ resulting in a sharp trade-off with consistency, which crashes to 0.34. By contrast, our method injects randomness directly into the typological feature selection (e.g., randomly sampling relative order of subject, object, and verb), allowing the language model to generate the description at a lower temperature ($T = 0.6$) to ensure logically coherent and fluent text. This maintains higher internal consistency (0.43) while achieving high typological diversity (0.60). We further note that alternative diversity-enhancing methods, such as specialized decoding strategies (Hewitt et al., 2022; Chang et al., 2025) or internal interventions (Chung et al., 2025; Zhou et al., 2025), typically require access to model logits or gradients. These are not feasible in our setting as we focus on state-of-the-art large reasoning models, which are either closed-weights (e.g., Gemini-2.5-pro) or prohibitively large to run locally (e.g. DeepSeek-R1).

3.5 Qualitative Analysis and Application

Beyond metrics, we examine the creative and linguistic quality of generations. Table 4 illustrates unconditional ConlangCrafter generations. The incorporation of unique typological features demonstrates the system's grasp of cross-linguistic variation beyond major world languages, and its effectiveness at diversity between generations. Table 4 also includes two examples of languages gen-

erated with user-input constraints, demonstrating user control over the generation process. These include an all-vowel language in which there are no consonant phonemes, and a language designed for an alien cephalopod species in which sound-based phonology is entirely replaced by color-based "chromemes" and gesture-based "kinemes." These examples illustrate that ConlangCrafter can accommodate creative and speculative constraints far beyond the typological space of natural languages, enabling applications in world-building and theoretical linguistic exploration. Additional controlled examples are provided in the appendix (Table 6). Along with these promising qualitative results, we note that quantitative evaluation of faithfulness to user-input constraints remains an open challenge, as they are inherently open-ended, variable in scope, and lack a definitive ground truth for automatic evaluation.

We provide a HTML interface to view these and other generated languages on our project page.

4 Related Work

Computational Conlanging. While several software tools[3] have been created by the conlanging community to automate various aspects of conlanging, machine learning-based methods are limited and piecemeal. Prior works have explored generating words in isolation (Zacharias et al., 2022), assigning or interpreting visual associations with nonsense words (Alper and Averbuch-Elor, 2023; Matsuhira et al., 2024; Kouwenhoven et al., 2025), and the ability of neural models to learn constructed languages with controlled linguistic properties (McCoy et al., 2018, 2021; Kallini et al., 2024; McCoy and Griffiths, 2025) or constructed via cryptographic substitutions (Marmonier et al., 2025). Recent work has also studied statistical properties of generated text when prompting LLMs to construct languages (Marmonier et al., 2025), but this is done in a single prompt and does not output fine-grained interpretable linguistic structure. By contrast, our approach constructs languages end-to-end with LLMs, with linguistically interpretable structure enabling downstream translation.

Computational Creativity. Generative models are increasingly being leveraged for creative tasks, with recent works exploring the use of LLMs for applications such as story generation (Venkatraman et al., 2025; Teleki et al., 2025), collaborative narrative modeling (Qiu and Hu, 2025), and research ideation (Si et al., 2025). Besides fully autonomous systems, human-computer interaction studies have explored the use of generative models as a creativity aid (Chakrabarty et al., 2024; Kumar et al., 2025). Similarly to our work, some studies have explored the invention of novel words (Malkin et al., 2021) and visual concepts (Richardson et al., 2024). A key challenge is evaluating performance on tasks which are inherently open-ended, which has motivated various evaluation frameworks and benchmarks for creativity (Li et al., 2025; Hou et al., 2025; Zhao et al., 2025). In this vein, we address a novel creative domain that requires both typological diversity and strict logical consistency, distinct from typical creative generation tasks.

Diverse Text Generation. LLMs often suffer from limited diversity, as common decoding strategies fail to match human text statistics (Holtzman et al., 2020), potentially exacerbated by post-training (Kirk et al., 2023; Yun et al., 2025). Mitigation strategies include temperature, nucleus (Holtzman et al., 2020), and min-p sampling (Nguyen et al., 2024), as well as handcrafted prompting techniques (Tian et al., 2024; Wang et al., 2025; Hu et al., 2025). These address token-level or semantic diversity, while we require targeted structural diversity at the level of linguistic typology.

5 Conclusion

We have proposed the novel paradigm of computational conlanging, showing that ConlangCrafter can construct coherent artificial languages through a novel multi-hop pipeline, validated by a new scalable evaluation framework for this task as well as manual expert judgements. ConlangCrafter offers a new computational creativity tool for language construction, enabling user-guided conlang creation with potential future applications such as procedural society generation and distillation of meta-linguistic reasoning for low-resource language NLP. We foresee extensions such as scaling to larger grammars and lexicons, exploring additional language aspects (e.g. semantics, multi-modality), extending the set of models tested to create a comprehensive benchmark, and modeling languages as evolving communication tools between agents across time, space, and culture.

[3] Such as those listed at the FrathWiki

Limitations

While our pipeline is designed to encourage diversity and avoid collapse to existing languages, LLMs may still be biased towards English and other high-resource languages in ways not captured by our evaluation framework, paralleling known issues with LLMs applied in multilingual settings (Chen et al., 2024; Singh et al., 2025). The limited nature of linguistic typological information seen during training (and the limited coverage of the worlds' languages in the linguistic literature overall) may prevent our method from generating extremely unusual features. Moreover, LLMs cannot capture the full range of human creativity and may lack the insight to invent novel, higher-level philosophical ideas.

Our language summaries only describe fundamental language components (phonology, morphosyntax, lexicon) while disregarding aspects such as semantics, pragmatics, discourse strategies, and orthography, which could be added in future work. These summaries are too short to capture a fraction of the complexity occurring in real languages; future research could study how to scale up their size, which currently is incompatible with our prompting methods which are limited by the context length of current LLMs.

Our evaluation method is limited due to the extensive labor required for manual evaluation, the potential for annotator priming from displayed automatic judgments (see Section 3.2.3), the open challenge of quantifying adherence to user-input constraints (Section 3.5), and the overall difficulty in formulating automated metrics for conlanging. Future work could design more scalable evaluation frameworks for this task. Additionally, our pipeline incurs a substantial computational cost due to repeated LLM calls with long contexts, particularly in self-refinement loops; future work could distill ConlangCrafter to achieve similar results with efficient inference.

Ethical Considerations

As with other uses of LLMs and other generative models, our method requires responsible use and disclaimers accompanying generated content to avoid disseminating misinformation. This includes concerns such as potential use of generated conlangs to evade content moderation, and possible cultural misrepresentation when generating languages inspired by real-world cultures. Repeated calls to LLM generation may consume significant compute resources, and we look to future work to improve the efficiency of our method. Finally, while our work may have future applications to low-resource languages, we emphasize that generation of fictional content must not come at the expense of research attention and resources being directed toward living communities.

Regarding manual annotation, no personally identifying information or sensitive data were collected, and only aggregated results are reported. We did not obtain IRB approval because the research does not involve human subjects. Individuals' participation was limited to professional linguistic annotation, and the research questions concern language model behavior, not the annotators.

Acknowledgments

We thank Alexander Elias, Allegra Robertson Molinaro, Kai Schenk, and Wesley Kuhron Jones for their linguistic assistance. We also thank Eric Chen and Hanzhi Zhu for their helpful feedback.

References

2013. The world atlas of language structures online (wals). Max Planck Institute for Evolutionary Anthropology [Data set]. Version v2020.4; accessed 2025-08-02 at https://wals.info.

Morris Alper and Hadar Averbuch-Elor. 2023. Kiki or bouba? sound symbolism in vision-and-language models. *Advances in Neural Information Processing Systems*, 36:78347–78359.

Jacopo Amidei, Paul Piwek, and Alistair Willis. 2018. Rethinking the agreement in human evaluation tasks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3318–3329.

Gasper Begus, Maksymilian Dabkowski, and Ryan Rhodes. 2025. Large linguistic models: Investigating llms' metalinguistic abilities. *IEEE Transactions on Artificial Intelligence*.

Tuhin Chakrabarty, Vishakh Padmakumar, Faeze Brahman, and Smaranda Muresan. 2024. Creativity support in the age of large language models: An empirical study involving professional writers. In *Proceedings of the 16th Conference on Creativity & Cognition*, pages 132–155.

Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. 2025. Real sampling: Boosting factuality and diversity of open-ended generation by extrapolating the entropy of an infinitely large lm. *Transactions of the Association for Computational Linguistics*, 13:760–783.

Pinzhen Chen, Simon Yu, Zhicheng Guo, and Barry Haddow. 2024. Is it good data for multilingual instruction tuning or just bad multilingual evaluation for large language models? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9706–9726.

John Joon Young Chung, Vishakh Padmakumar, Melissa Roemmele, Yuqian Sun, and Max Kreminski. 2025. Modifying large language model post-training for diverse creative writing. *arXiv preprint arXiv:2503.17126*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Carol Genetti. 2018. *Introduction: Language, Languages, and Linguistics*. Cambridge University Press.

Michael Ginn, Mans Hulden, and Alexis Palmer. 2024a. Can we teach language models to gloss endangered languages? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5861–5876.

Michael Ginn, Ali Marashian, Bhargav Shandilya, Claire Post, Enora Rice, Juan Vásquez, Marie Mcgregor, Matthew Buchholz, Mans Hulden, and Alexis Palmer. 2024b. On the robustness of neural models for full sentence transformation. In *Proceedings of the 4th Workshop on Natural Language Processing for Indigenous Languages of the Americas (AmericasNLP 2024)*, pages 159–173.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, and 1 others. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Aspen K Hopkins and Alex Renda. 2023. Can llms generate random numbers? evaluating llm sampling in controlled domains. Sampling and Optimization in Discrete Space (SODS) ICML 2023 Workshop.

Zhaoyi Joey Hou, Bowei Alvin Zhang, Yining Lu, Bhiman Kumar Baghel, Anneliese Brei, Ximing Lu, Meng Jiang, Faeze Brahman, Snigdha Chaturvedi, Haw-Shiuan Chang, and 1 others. 2025. Creativityprism: A holistic benchmark for large language model creativity. *arXiv preprint arXiv:2510.20091*.

Wenyang Hu, Gregory Kang Ruey Lau, Liu Diwen, Chen Jizhuo, See Kiong Ng, and Bryan Kian Hsiang Low. 2025. Dipper: Diversity in prompts for producing large language model ensembles in reasoning tasks. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 35546–35560.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2022. Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*.

Tom Kouwenhoven, Max Peeperkorn, Roy De Kleijn, and Tessa Verhoef. 2025. Shaping shared languages: Human and large language models' inductive biases in emergent communication. *arXiv preprint arXiv:2503.04395*.

Harsh Kumar, Jonathan Vincentius, Ewan Jordan, and Ashton Anderson. 2025. Human creativity in the age of llms: Randomized experiments on divergent and convergent thinking. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–18.

Ruizhe Li, Chiwei Zhu, Benfeng Xu, Xiaorui Wang, and Zhendong Mao. 2025. Automated creativity evaluation for large language models: A reference-based approach. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21475–21488, Suzhou, China. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36:46534–46594.

Nikolay Malkin, Sameera Lanka, Pranav Goel, Sudha Rao, and Nebojsa Jojic. 2021. Gpt perdetry test: Generating new meanings for new words. In *2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5542–5553. Association for Computational Linguistics.

Malik Marmonier, Rachel Bawden, and Benoît Sagot. 2025. Explicit learning and the llm in machine translation. *arXiv preprint arXiv:2503.09454*.

Chihaya Matsuhira, Marc A Kastner, Takahiro Komamizu, Takatsugu Hirayama, and Ichiro Ide. 2024. Investigating conceptual blending of a diffusion model for improving nonword-to-image generation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 7307–7315.

R Thomas McCoy and Thomas L Griffiths. 2025. Modeling rapid language learning by distilling bayesian priors into artificial neural networks. *Nature Communications*, 16(1):1–14.

Richard Thomas McCoy, Jennifer Culbertson, Paul Smolensky, and Géraldine Legendre. 2021. Infinite use of finite means? evaluating the generalization of center embedding learned from an artificial grammar. In *Proceedings of the annual meeting of the cognitive science society*, volume 43.

Thomas R McCoy, Robert Frank, and Tal Linzen. 2018. Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 40.

Minh Nhat Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2024. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*.

Ziliang Qiu and Renfen Hu. 2025. Deep associations, high creativity: A simple yet effective metric for evaluating large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10870–10883.

Elad Richardson, Kfir Goldberg, Yuval Alaluf, and Daniel Cohen-Or. 2024. Conceptlab: Creative concept generation using vlm-guided diffusion prior constraints. *ACM Transactions on Graphics*, 43(3):1–14.

Christine Schreyer. 2021. Constructed languages. *Annual Review of Anthropology*, 50(1):327–344.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. In *The Thirteenth International Conference on Learning Representations*.

Toby Simonds, Kevin Lopez, Akira Yoshiyama, and Dominique Garmier. 2025. Self rewarding self improving. *arXiv preprint arXiv:2505.08827*.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, and 1 others. 2025. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799.

Garrett Tanzer, Mirac Suzgun, Eline Visser, Dan Jurafsky, and Luke Melas-Kyriazi. 2023. A benchmark for learning to translate a new language from one grammar book. *arXiv preprint arXiv:2309.16575*.

Maria Teleki, Vedangi Bengali, Xiangjue Dong, Sai Tejas Janjur, Haoran Liu, Tian Liu, Cong Wang, Ting Liu, Yin Zhang, Frank Shipman, and 1 others. 2025. A survey on llms for story generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13954–13966.

Yufei Tian, Abhilasha Ravichander, Lianhui Qin, Ronan Le Bras, Raja Marjieh, Nanyun Peng, Yejin Choi, Thomas L Griffiths, and Faeze Brahman. 2024. Macgyver: Are large language models creative problem solvers? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5303–5324.

Chris Van der Lee, Albert Gatt, Emiel Van Miltenburg, and Emiel Krahmer. 2021. Human evaluation of automatically generated text: Current trends and best practice guidelines. *Computer Speech & Language*, 67:101151.

Saranya Venkatraman, Nafis Irtiza Tripto, and Dongwon Lee. 2025. Collabstory: Multi-llm collaborative story generation and authorship analysis. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3665–3679.

Eline Visser. 2022. *A grammar of Kalamang*. Language Science Press.

Qihan Wang, Shidong Pan, Tal Linzen, and Emily Black. 2025. Multilingual prompting for improving LLM generation diversity. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 6378–6400, Suzhou, China. Association for Computational Linguistics.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*.

Longfei Yun, Chenyang An, Zilong Wang, Letian Peng, and Jingbo Shang. 2025. The price of format: Diversity collapse in llms. *arXiv preprint arXiv:2505.18949*.

Thomas Zacharias, Ashutosh Taklikar, and Raja Giryes. 2022. Extending the vocabulary of fictional languages using neural networks. *arXiv preprint arXiv:2201.07288*.

Chen Zhang, Mingxu Tao, Quzhe Huang, Zhibin Chen, and Yansong Feng. Can llms learn a new language on the fly? a case study on zhuang. In *The Second Tiny Papers Track at ICLR 2024*.

Kexun Zhang, Yee Man Choi, Zhenqiao Song, Taiqi He, William Yang Wang, and Lei Li. 2024. Hire a linguist!: Learning endangered languages with in-context linguistic descriptions. *arXiv preprint arXiv:2402.18025*.

Yunpu Zhao, Rui Zhang, Wenyi Li, and Ling Li. 2025. Assessing and understanding creativity in large language models. *Machine Intelligence Research*, 22(3):417–436.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Kuan Lok Zhou, Jiayi Chen, Siddharth Suresh, Reuben Narad, Timothy T. Rogers, Lalit K Jain, Robert D Nowak, Bob Mankoff, and Jifan Zhang. 2025. Bridging the creativity understanding gap: Small-scale human alignment enables expert-level humor ranking in LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16273–16287, Suzhou, China. Association for Computational Linguistics.

A Implementation Details

A.1 Pipeline Design and Prompts

Full prompts used in our pipeline are provided in Figures 4 to 9. In each prompt, fields in curly braces { } are values that are filled in at inference time; these include configuration variable (e.g. `{n_questions}` has default value 10) as well as outputs of previous steps (e.g. `{checklist}` in the second prompt in Figure 4 is filled in with the output from the previous prompt).

Unless otherwise specified, the prompts in Figures 4 to 9 are run serially. The last lexicon prompt (bottom of Figure 6) is run in a loop until the lexicon contains a minimum number of items (100 by default). The self-refinement prompts (Figures 7 to 8) are run after corresponding steps; the critic prompts there are used to identify issues, and the amendment prompts edit to address these issues. These are run in a loop until the critic returns an overall score above a fixed threshold.

We proceed to describe core design choices motivating this pipeline, beyond the overall multi-hop structure, randomness injection, and self-refinement components described in the main paper. Figure 4 includes a third step discussing word shapes to encourage diverse word shapes as we find that otherwise languages are prone to collapsing to monosyllables (typologically attested but limiting diversity of generations). Figure 5 includes an expansion step to achieve a more satisfying level of detail in the output grammar, with an additional merger step to holistically incorporate these details into the grammar. Many prompts explicitly request interesting and unusual outputs, which complements our goal of constructing diverse and interesting languages.

A.2 Baseline Prompt

The single prompt used for baseline comparisons is given in Figure 10. This uses the same output fields as the full ConlangCrafter pipeline, while requesting an entire language sketch and all sentence translations in a single prompt.

A.3 Evaluation Prompts

The prompts used for evaluation metrics are shown in Figures 11 to 12. This includes the prompt used for WALS feature extraction and the prompt used for the judge evaluating translation consistency.

A.4 Constructive Translation

For constructive translation, we translate the following fixed sentences which include a variety of syntactic structures as detailed below:

- The big dog is sleeping. *(Attributive adjective, present continuous, definite article)*
- Where is my book? *(WH-interrogative, possessive, copula)*
- She will give him water. *(Ditransitive, future tense, pronouns)*
- This child walked to that house yesterday. *(Demonstratives, past tense, directional, temporal adverb)*
- Are you hungry? *(Yes/no question, adjectival predicate)*
- Give me the red bird! *(Imperative, indirect object pronoun, attributive adjective)*

WALS#	Feature	Description/Options
1	Consonant inventory	Small (≤ 20) vs. Large (> 20)
2	Vowel inventory	Large (≥ 9) vs. Small (< 9)
13	Tone	Tonal vs. non-tonal systems
20	Morphological fusion	Isolating, agglutinating, fusional, etc.
26	Affixation balance	Relative preference for prefixing vs. suffixing
30	Gender inventory	Number of gender distinctions (including none)
49	Case inventory	Minimal, moderate, or extensive case systems
69	Numeral classifiers	Classifier vs. non-classifier systems
81	Basic word order	SVO, SOV, VSO, etc.
85	Adposition type	Prepositions vs. postpositions
86	Genitive–noun order	Genitive before vs. after noun
87	Adjective–noun order	Adjectives precede or follow nouns
90	Relative-clause order	Head-initial vs. head-final
98	Alignment typology	Nominative–accusative vs. active–stative
107–111	Valence morphology	Presence/absence of causative, passive, applicative
116	Question-marking strategy	Interrogative particle vs. inversion

Table 5: Linguistic features for typological analysis, with WALS numbers (WALS#) listed.

- The woman and the man are talking. *(Coordination, present continuous)*
- I do not see three cats. *(Negation, transitive, numeral, plural)*
- There is a black mountain. *(Existential construction, indefinite article, attributive adjective)*
- Two children played in the garden. *(Numeral, past tense, locative prepositional phrase)*

New lexical items and grammar may be appended to the language sketch, though in our evaluations we translate each sentence independently (without re-adding new lexical items and grammar rules to the language sketch).

A.5 WALS Features and Visualization

WALS features that are extracted and used for typological diversity calculations are shown in Table 5. We include their numbers from the WALS database, along with an overall description of the contents of each feature. The visualization in Figure 3 uses default scikit-learn t-SNE hyperparameters.

A.6 Language Generation Details

LLM generation uses the maximum values for maximum output tokens and otherwise using default recommended hyperparameters for decoding (temperature, top-p). For metric calculations in our experiments, we sample $N \approx 20$ languages.

Due to resource constraints, the number of languages generated for metric calculations varies depending on setting (16 for Gemini-2.5-Flash, 20 for Gemini-2.5-Pro, and 24 for DeepSeek-R1). We use rejection sampling to remove degenerate outputs that do not conform to the required language sketch format, rejecting about 5-10% of the samples, depending on the model.

For self-refinement, the critic provides an overall score on a 1–10 scale, with 9 being explicitly given as the score for content which is fully consistent but may contain minor unclear points. Self-refinement terminates when this score exceeds a fixed threshold (9 for language sketch generation, 10 for translation), or after 10 iterations.

B Additional Results

B.1 Example Language Sketches

On our project page, we provide various ConlangCrafter-generated language sketches and translations as a browsable HTML page. In addition, we provide the full text of a single language sketch and constructive translation outputs in Figures 13 to 19. Note that the sketch is saved as free text. In practice, the phonology and grammar sections are markdown-formatted, while the lexicon is in CSV format.

User-Input Control	Sample Sentence	Notes
The language is a creole combining Japanese and Esperanto.	matːa-o amai pãno-un ʔamiko wait-NPST sweet bread-ACC friend “A friend waits for the sweet bread.”	Cf. Esperanto *amiko* “friend”, *pano*, “bread”; Japanese *matta* “waited”, *amai* “sweet”, *pan* “bread”
This language is nasal-centric, with all consonants and vowels nasalized and a rich vocabulary and grammar centered around smells.	^{n}‖ə̃m-^{ŋ}gə̃ má-^{n}|ɔːn-ŋàn sleep-3SG.I.AGR I.SG-person-NOM “The person is sleeping (I can smell their sleeping scent).”	Smell-based grammar includes smell-based noun classes and olofactory evidentiality (for events experienced by smell).
All inflectional forms are suppletive, and words in a sentence are ordered alphabetically.	janta kasa sake see[3S.S/3S.O,INFER] person.ACC person.NOM “The person sees the person (I infer).”	Example sentence shows suppletion (of “person”) and is in alphabetical order: j < k < s. Other suppletive forms of “to see” include fans (see[3S.S/3S.O,VIS]), teso (seeing.NMLZ), pote (see[3S.S/3S.O,REPOR]), and more.
Verbless constructions (null verb) are common and have a variety of meanings depending on the argument frame present, being used for many functions that are typically lexical verbs (far beyond copular functions).	st^{w}ũ k^{j}o.ne mẽ.l^{j}a n̪a.p^{j}ĩ.na ɓa p^{w}jel^{j} PFV.PST ERG.III teacher(III) DAT.III person(III) water(IV) “The teacher made the person drink water.”	Lit. “(Did) teacher to person water”; st^{w}ũ is a TAM particle and the predicate “made drink” is implied.
Triple center embeddings are common.	p^{h}uːˈmor-wo=o [p^{h}esi-e=je [oˈq^{h}ɔs-om=jom t^{h}eˈk^{h}iːsir-na] jeˈkweːt^{h}is-na] ruˈp^{h}uŋ-om=om oˈxoː t͡s^{h}ɛn man-AGT=3A [dog-INTR=LOGO.S [rock-PAT=LOGO.P chase-CONV] see-CONV] food-PAT=3P FUT eat “The man, who sees his dog which is chasing his (the man’s) rock, will eat the food.”	From the grammar: “A defining syntactic feature is its routine use of deep center-embedding... The system allows for multiple, nested embeddings, with triple center-embeddings being common in narrative and formal speech.”

Table 6: **Qualitative examples** of ConlangCrafter-generated languages with user-input conditions, demonstrating user control and the ability to test creative, linguistic, and philosophical ideas.

B.2 Qualitative Results with User-Input Constraints

In Table 6, we show additional qualitative examples from languages generated with user-input constraints (beyond those presented in the main paper in Table 4), which our system accepts as free text. These illustrate that our system succeeds in generating creative conlangs that adhere to the user-input specifications, allowing users to test creative, linguistic, and philosophical ideas beyond what is attested in natural languages.

C Computation Usage

As our tests focus on large models run via external APIs, we use token counts as the primary measure of computation. This value varies depending on the model used and due to randomness across runs. When using DeepSeek-R1, our full pipeline, including self-refinement and sentence translations, consumes approximately 660K tokens for a single language, costing about 4 USD on the Together AI platform. This cost is largely driven by self-refinement loops; generating a single language sketch without self-refinement or sentence translations consumes approximately 70K tokens, representing a ten-fold decrease. By comparison, using Gemini 2.5 Flash, generating a complete language with our standard translation set requires an average of approximately 2.15M tokens. In this case, most of the cost is incurred during the translation stage and within the iterative self-refinement QA loops, which are critical for maintaining internal consistency.

D Annotator Instructions and Protocol

D.1 Manual Diversity Evaluation

Materials shown to annotators. Annotators were shown language sketches (phonology and grammar sections only) along with automatically-inferred WALS features. Each feature was shown with the chosen value highlighted, as well as all possible values for that feature.

Full Instructions Provided.

The browsable HTML file includes 20 generated languages. Each is shown with its description (phonology and grammar) followed by

automatically-inferred typological (WALS) features in the section "WALS Features for Diversity Calculation". Your task is to review these automatic features and validate whether they are correctly inferred from the language description.

In the spreadsheet, for each language and feature, please indicate one of the following values:

- *0: If the automatic feature value is a mistake (e.g. automatic value said SOV but language is actually SVO)*
- *1: If the automatic feature value is correct*
- *2: If the automatic feature value is undefined but should be defined (e.g. automatic value is empty but should be SVO)*
- *3: If the automatic feature value is defined but should be undefined (e.g. automatic value is SVO but word order is actually not defined anywhere in the language sketch)*

D.2 Manual Consistency Evaluation

Materials shown to annotators. Annotators were shown full language sketches along with constructive translation results. Each translation entry presented:

1. the source English sentence (prompt),
2. the generated target sentence (surface form),
3. an interlinear gloss,
4. optional fields documenting any *new lexical items* or *new grammatical rules* the generator introduced,
5. an automatically produced LLM judgement (inconsistent / mostly consistent / consistent) with a brief rationale, shown as a non-authoritative hint.

Full Instructions Provided. The following is the full text of the instructions provided for annotation: *[You are provided with] a browsable HTML file with 20 generated languages for the eval. Each language has its description (phonology and grammar) followed by the translations of ten sentences. This time, each sentence translation is shown with the result of an LLM judge which tries to decide if the translation is consistent with the language description – it judges it as either consistent, mostly consistent, or inconsistent, along with its reasoning. It may sometimes be wrong, but you can use it as a hint to try to decide if a translation is valid.*

[In the provided spreadsheet], for each sentence in each language, please enter 0 if the sentence is inconsistent with the language, 1 if it is mostly consistent, and 2 if it is fully consistent with the language.

Each translation has the following format:

- *Top, in italics: The English sentence to be translated.*
- *Bold: The attempted translation of this sentence into the language*
- *Typewriter font: The attempted interlinear gloss of this translation.*
- *"New Words": If the language description was missing some words required to translate the sentence, this should list new words that were invented for the language to be able to translate the sentence.*
- *"New Grammar Rules": If the language description was missing some grammar points needed to be able to translate the sentence, this lists these new invented grammar points ("rule"), along with explanations ("justification") for why they should make sense given the language's description.*
- *LLM judgement – inconsistent / mostly consistent / consistent + explanation. As stated above, these are not guaranteed to be correct, but may help you decide if a translation is really consistent with the language or not.*

Important guidelines:

- *Consider a translation valid as long as it is consistent with the language's description, even if new words and/or grammar points had to be invented to translate it.*
- *Different translations of different sentences might make different decisions for new words and grammar. They do not need to be consistent with each other – only judge if a single translation is consistent with the language's original description.*

```
I am designing a hypothetical language's phonology. Provide me with {n_questions} sliders with scales from 1 to {scale_size}, and/or multiple-choice questions with {n_answers} possible answers (numbered 1-{scale_size}), to determine the most important typological features to construct this language's phonology. There should be {n_questions} of them total (counting sliders and multiple-choice questions together).
Provide no additional explanation or discussion.
```

```
I am designing a hypothetical language's phonology.
Consider the following checklist:
== START CHECKLIST ==
{checklist}
== END CHECKLIST ==
Use values {values} for those respectively.
It should also obey the following constraint(s); if these contradict the above, these take priority:
{custom}
Now write a summary of its phonology.
Make sure there are a couple of aspects that are unusual, creative, interesting, or surprising, while still obeying the constraints from above.
Format your output as follows. Give no additional explanation or discussion.
# Phonology
## Consonants
(IPA chart of consonants, as markdown table)
## Vowels
(IPA chart of vowels, as markdown table)
## Phonotactics
(Brief, single-paragraph explanation)
## Suprasegmentals
(Brief, single-paragraph explanation)
```

```
A hypothetical language has the following phonology:
=== START ===
{phonology}
=== END ===
Write a description of the distribution of word shapes in this language, touching content vs. function items and the distribution of word lengths and syllable counts. Use qualitative descriptors (like "most," "many," "some," "few," "rare") rather than specific percentages or numbers.
Include at least {n} diverse lexical items that illustrate the points discussed.
Give the words in IPA (underlying phonemic representation only), making sure all phonemic features are indicated, including contrastive suprasegmentals (if relevant). Do not give their translations in English; only note if they are content or function items and how common or uncommon they are.
The new lexical items should obey the following constraints:
{custom}
If these contradict anything above, these constraints take priority.
Format your output as follows. Give no additional explanation or discussion.
## Word Shapes and Lexical Statistics
(description here)
```

Figure 4: Phonology prompts.

```
I am designing a hypothetical language's grammar (morphology and syntax). Provide me with {n_questions} sliders with scales from 1 to {scale_size}, and/or multiple-choice questions with {n_answers} possible answers (numbered 1-{scale_size}), to determine the most important typological features to construct this language's grammar (morphology and syntax). There should be {n_questions} of them total (counting sliders and multiple-choice questions together).
Provide no additional explanation or discussion.
```

```
I am designing a hypothetical language's grammar (morphology and syntax).
Consider the following checklist:
== START CHECKLIST ==
{checklist}
== END CHECKLIST ==
Use values {values} for those respectively.
It should also obey the following constraint(s); if these contradict the above, these take priority:
{custom}
The language has the following phonology:
=== START ===
{phonology}
=== END ===
Now write a summary of its grammar (morphology and syntax).
Make sure there are a couple of aspects that are unusual, creative, interesting, or surprising, while still obeying the constraints from above.
For any statement, include examples with interlinear glosses and English translations. When possible, use words/roots from the list above in examples. Examples should be provided right next to the statements that they illustrate (not all at the end).
Format your output as follows. Give no additional explanation or discussion.
# Grammar
## Morphology
(summary of morphology here)
## Syntax
(summary of syntax here)
```

```
A hypothetical language has the following phonology:
=== START ===
{phonology}
=== END ===
Here is a summary of its grammar (morphology and syntax):
=== START ===
{grammar}
=== END ===
Write expanded grammar sections including important points that are missing above, that would be needed to understand or use the language.
Make sure there are a couple of aspects that are unusual, creative, interesting, or surprising, while still obeying the constraints from above.
It should also obey the following constraint(s); if these contradict the above, these take priority:
{custom}
For any statement, include examples with interlinear glosses and English translations. When possible, use words/roots from the list above in examples. Examples should be provided right next to the statements that they illustrate (not all at the end).
```

```
Combine the summaries below together. If there are any contradictions, amend as needed to make everything consistent. Make sure to include all information and examples that appear anywhere in any of the summaries.
Only return the new summary (do not give any further commentary or explain your amendations).
{summaries}
Format your output as follows. Make sure it contains all information from the summaries above. Examples include both interlinear glosses and English translations, and should be provided right next to the statements that they illustrate (not all at the end). Give no additional explanation or discussion.
# Grammar
## Morphology
(morphological information here)
## Syntax
(syntactic information here)
```

Figure 5: Grammar prompts.

```
 A hypothetical language has the following phonology and grammar:
=== START ===
{phonology}
{grammar}
=== END ===
Output a CSV with a lexicon of the language, including all lexical items that appear in the
language's information above.
The CSV column names and their contents should be:
* "ipa": Lexical item as IPA (without brackets)
* "pos": Part of speech
* "translation": Translation of item into English
* "grammar": Any grammar points or information specific to that lexical item (e.g. inflectional or
conjugational classes, irregularities, ...)
* "derivation": If the item is derived, note its derivation here
* "notes": Any additional notes
Provide translations for all lexical items, even those whose translations are not explicitly
specified in the provided background above (i.e. don't list any translations as "unknown").
Do not provide any further explanation or text, only the CSV.
```

```
 A hypothetical language has the following phonology:
=== START ===
{phonology}
=== END ===
It has the following grammar:
=== START ===
{grammar}
=== END ===
It has the following lexicon:
=== START ===
{lexicon}
=== END ===
Propose at least {n} additional lexical items to add to the lexicon, including a diverse set of
items covering common and uncommon concepts and different parts of speech. Make sure they obey
the language's phonology and morphology, including following the word shape distribution described
in "Word Shapes and Lexical Statistics". Also make sure they do not overlap any existing lexical
items.
Output a CSV in the same format as the lexicon above. The CSV column names and their contents should
be:
* "ipa": Lexical item as IPA (without brackets)
* "pos": Part of speech
* "translation": Translation of item into English
* "grammar": Any grammar points or information specific to that lexical item (e.g. inflectional or
conjugational classes, irregularities, ...)
* "derivation": If the item is derived, note its derivation here
* "notes": Any additional notes
Do not provide any further explanation or text, only the CSV.
```

Figure 6: Lexicon prompts.

```
 Here is a hypothetical language's {content_type}:
=== START ===
{content}
=== END ===
Is the description of the language's {content_type} consistent? Return a JSON with keys:
* "overall_score": Integer from 1 (completely inconsistent) to 10 (completely consistent), scoring
the {content_type} overall. If there are not issues that are strictly contradictions or errors (for
example, if the only issues are ambiguities/unclear points), the score should be 9 (consistent but
somewhat unclear). If there are only minor contradictions or errors, the score should be 8 (very
minor inconsistencies).
* "issues": List of errors and inconsistencies found (if any) and their corrections. Each item in
this list should be an object with the following keys and values:
* "issue": String describing issue
* "type": One of "inconsistency"/"error"/"ambiguity", indicating if the issue is an inconsistency
(contradiction), error (other mistake), or ambiguity (not strictly a mistake but unclear)
* "correction": String describing how it should be corrected
* "priority": 1 (high − severe issue), 2 (medium), or 3 (low − minor or very minor issue)
Be as exhaustive as possible in your search for issues.
Output only this JSON object. Give no additional explanation or discussion.
```

```
 Here is a hypothetical language's {content_type}:
=== START ===
{content}
=== END ===
For context, here is the language's {context_type}:
=== START ===
{context}
=== END ===
Is the description of the language's {content_type} consistent with itself and with its
{context_type}? Return a JSON with keys:
* "overall_score": Integer from 1 (completely inconsistent) to 10 (completely consistent), scoring
the {content_type} overall. If there are not issues that are strictly contradictions or errors (for
example, if the only issues are ambiguities/unclear points), the score should be 9 (consistent but
somewhat unclear). If there are only minor contradictions or errors, the score should be 8 (very
minor inconsistencies).
* "issues": List of errors and inconsistencies found (if any) and their corrections. Each item in
this list should be an object with the following keys and values:
* "issue": String describing issue
* "type": One of "inconsistency"/"error"/"ambiguity", indicating if the issue is an inconsistency
(contradiction), error (other mistake), or ambiguity (not strictly a mistake but unclear)
* "correction": String describing how it should be corrected in the {content_type}. Note that only
the {content_type} can be corrected, not the {context_type}. If the issue is due to a conflict with
the {context_type}, make sure this description will be clear to an editor who does not have the
{context_type} in front of them when amending the {content_type}.
* "priority": 1 (high − severe issue), 2 (medium), or 3 (low − minor or very minor issue)
Be as exhaustive as possible in your search for issues.
Output only this JSON object. Give no additional explanation or discussion.
```

```
 Here is a hypothetical language's phonology:
=== START ===
{content}
=== END ===
Here is a judgement of its overall consistency on a scale of 1 (completely inconsistent) - 5
(completely consistent), and specific issues found (if any) and their priorities (from 1=high to
3=low) and how to correct them:
{judgement}
Correct these points (and any other errors or inconsistencies, if any), outputting an amended
version (without === START === / === END === lines). Give no additional explanation or discussion.
```

Figure 7: Self-refinement prompts.

```
 Here is a translation result for a constructed language:
=== START ===
{content}
=== END ===
For context, here is the language specification:
=== START ===
{context}
=== END ===
Evaluate the quality and accuracy of this translation. Return a JSON with keys:
* "overall_score": Integer from 1 (completely incorrect) to 10 (excellent translation), scoring the
translation overall. Consider:
- Adherence to phonological rules (10: fully consistent, 8-9: minor violations, 5-7: some violations,
1-4: major violations)
- Adherence to grammatical rules (10: fully consistent, 8-9: minor violations, 5-7: some violations,
1-4: major violations)
- Appropriate use of lexicon (10: excellent use, 8-9: mostly appropriate, 5-7: some issues, 1-4:
poor lexical choices)
- Quality of glossing (10: accurate and complete, 8-9: mostly accurate, 5-7: some errors, 1-4: poor
glossing)
- Semantic accuracy (10: meaning preserved, 8-9: mostly preserved, 5-7: some meaning lost, 1-4:
meaning significantly altered)
* "issues": List of problems found (if any). Each item should be an object with:
* "issue": String describing the problem
* "type": One of "phonological_violation"/"grammatical_violation"/"lexical_error"/"glossing_error"/
"semantic_error"/"consistency_error"
* "correction": String describing how the translation should be corrected
* "priority": 1 (high - major error affecting meaning or violating core rules), 2 (medium -
noticeable error), or 3 (low - minor issue or style preference)
Be thorough in checking:
1. Phonological consistency: Do invented words follow the sound patterns and constraints?
2. Grammatical consistency: Are word order, morphology, and syntax rules followed?
3. Lexical appropriateness: Are existing words used correctly? Are new words justified and
well-formed?
4. Glossing accuracy: Does the gloss correctly break down morphemes and grammatical functions?
5. Semantic preservation: Does the translation convey the intended meaning?
6. Internal consistency: Are new rules and words consistent with each other?
Output only this JSON object. Give no additional explanation or discussion.
```

```
 Here is a translation result for a constructed language that might need improvement:
=== START ===
{content}
=== END ===
Based on this quality assessment:
=== START ===
{judgement}
=== END ===
Please provide an improved version of the translation that addresses the identified issues. Make
sure to:
1. Fix any phonological violations by adjusting words to follow the sound patterns
2. Correct grammatical errors to match the specified rules
3. Improve lexical choices and justify any new words created
4. Ensure accurate glossing with proper morpheme breakdown
5. Preserve semantic meaning while fixing technical issues
6. Maintain consistency across all elements
Return the corrected translation in the same JSON format as the original:
{{
"sentences": [
{{
"conlang_sentence": "<corrected sentence in the constructed language only - no glosses, morpheme
breaks, or explanations>",
"gloss": "<corrected word-by-word breakdown with morphemes and grammatical functions>",
"new_words": [{{"<new_word>": "<english_translation>"}}],
"new_grammar_rules":         [{{"rule":         "<description_of_new_rule>",      "justification":
"<explanation_of_why_needed_and_consistency>"}}]
}}
]
}}
Output only this JSON object. Give no additional explanation or discussion.
```

Figure 8: Self-refinement prompts (translation).

```
 You are a skilled linguist working with a constructed language. You have been provided with a
complete language specification including vocabulary, grammar and phonology.
A hypothetical language has the following phonology:
=== START ===
{phonology}
=== END ===
It has the following grammar:
=== START ===
{grammar}
=== END ===
{lexicon_section}
Your task is to translate the following English sentence into this constructed language:
English sentence: {input_sentence}
Instructions:
1. Use the vocabulary provided in the language specification (if available)
2. Follow the grammatical rules and patterns described
3. Apply the phonological conventions
4. If a word is not in the vocabulary, invent a new word that is consistent with the specifications
5. If a grammatical construction is needed but not specified, invent a new grammar rule that is
consistent with the existing language patterns
6. Ensure the translation follows the word order and morphological patterns specified
If you create new words, ensure they:
- Follow the phonological constraints of the language
- Have appropriate semantic scope for the meaning needed
- Use appropriate derivational processes if applicable
If you create new grammar rules, ensure they:
- Are consistent with existing grammatical patterns
- Follow the language's morphological and syntactic tendencies
- Are plausible extensions of the existing system
Provide a gloss line: break down the conlang sentence word by word, showing morphemes and their
grammatical functions using standard linguistic abbreviations (e.g., FUT, SBJ, OBJ, ACC, etc.). The
gloss should use English abbreviations and morpheme breakdowns, NOT the conlang words.
Return the result in JSON format only:
{{
"sentences": [
{{
"conlang_sentence": "<sentence in the constructed language>",
"gloss": "<word-by-word breakdown with morphemes and grammatical functions>",
"new_words": [{{"<new_word>": "<english_translation>"}}],
"new_grammar_rules":         [{{"rule":         "<description_of_new_rule>",      "justification":
"<explanation_of_why_needed_and_consistency>"}}]
}}
]
}}
```

Figure 9: Translation prompt.

```
 Create a conlang (constructed language). Include a description of its phonology, grammar (morphology
and syntax), and lexicon. Use the following format:
# Phonology
## Consonants
(IPA chart of consonants, as markdown table)
## Vowels
(IPA chart of vowels, as markdown table)
## Phonotactics
(Brief, single-paragraph explanation)
## Suprasegmentals
(Brief, single-paragraph explanation)
## Word Shapes and Lexical Statistics
(description here)
# Grammar
## Morphology
(morphological information here in markdown format)
## Syntax
(syntactic information here)
# Lexicon
(list of at least 100 lexical items in the language in a csv format)
# Corpus
give translations and interlinear glosses for the following 10 sentences.  Invent new words or
grammar rules, if needed, for each sentence.
Provide a gloss line: break down the conlang sentence word by word, showing morphemes and their
grammatical functions using standard linguistic abbreviations (e.g., FUT, SBJ, OBJ, ACC, etc.). The
gloss should use English abbreviations and morpheme breakdowns, NOT the conlang words.
1. The big dog is sleeping.
2. Where is my book?
3. She will give him water.
4. This child walked to that house yesterday
5. Are you hungry?
6. Give me the red bird!
7. The woman and the man are talking.
8. I do not see three cats.
9. There is a black mountain.
10. The two children played in the garden.
Return the translation results in JSON format:
{{
"sentences": [
{{
"english_sentence": "<original English sentence 1>",
"conlang_sentence": "<sentence in the constructed language>",
"gloss": "<word-by-word breakdown with morphemes and grammatical functions>",
"new_words": [{{"<new_word>": "<english_translation>"}}],
"new_grammar_rules":         [{{"rule":         "<description_of_new_rule>",      "justification":
"<explanation_of_why_needed_and_consistency>"}}]
}},
{{
"english_sentence": "<original English sentence 2>",
"conlang_sentence": "<sentence in the constructed language>",
"gloss": "<word-by-word breakdown with morphemes and grammatical functions>",
"new_words": [{{"<new_word>": "<english_translation>"}}],
"new_grammar_rules":         [{{"rule":         "<description_of_new_rule>",      "justification":
"<explanation_of_why_needed_and_consistency>"}}]
}}
]
}}
```

Figure 10: Baseline single prompt.

```
 Please analyze the following documentation for the constructed language "{your_language_name}" and return the typological analysis as a JSON object.
A hypothetical language has the following phonology:
=== START ===
{phonology}
=== END ===
It has the following grammar:
=== START ===
{grammar}
=== END ===
It has the following lexicon:
=== START ===
{lexicon}
=== END ===
## JSON Schema to Populate
{
"language_name": "{your_language_name}",
"analysis_timestamp_utc": "{current_utc_iso_timestamp}",
"typology": {
"wals_81A_svo_order": {
"value": "SVO | SOV | VSO | VOS | OVS | OSV | No Dominant Order | null",
"confidence": "High | Medium | Low"
},
"wals_85A_adpositions": {
"value": "Prepositions | Postpositions | Inpositions | No Adpositions | null",
"confidence": "High | Medium | Low"
},
"wals_87A_adjective_noun": {
"value": "Adjective-Noun | Noun-Adjective | null",
"confidence": "High | Medium | Low"
},
"wals_20A_fusion": {
"value": "Isolating | Agglutinative | Fusional | null",
"confidence": "High | Medium | Low"
},
"wals_26A_affixation": {
"value": "Strongly Suffixing | Weakly Suffixing | Equal Prefixing/Suffixing | Weakly Prefixing | Strongly Prefixing | null",
"confidence": "High | Medium | Low"
},
"wals_13A_tone": {
"value": "Tonal | Non-tonal | null",
"confidence": "High | Medium | Low"
},
"wals_30A_gender_count": {
"value": "None | Two | Three | Four | Five or more | null",
"confidence": "High | Medium | Low"
},
"wals_49A_case_count": {
"value": "None (0-1) | Minimal (2-3) | Moderate (4-5) | Extensive (6+) | null",
"confidence": "High | Medium | Low"
},
"wals_alignment": {
"value": "Accusative | Ergative | Active-Stative | Split | null",
"confidence": "High | Medium | Low"
},
"wals_87B_relative_clause_order": {
"value": "Head-initial | Head-final | Mixed | null",
"confidence": "High | Medium | Low"
},
"wals_genitive_order": {
"value": "Genitive-Noun | Noun-Genitive | null",
"confidence": "High | Medium | Low"
},
"wals_question_marking": {
"value": "Intonation | Particle | Morphological | Inversion | null",
"confidence": "High | Medium | Low"
},
"phoneme_inventory": {
"vowels": "Small (≤5) | Medium (6–8) | Large (≥9) | null",
"consonants": "Small (≤20) | Medium (21–35) | Large (≥36) | null",
"confidence": "High | Medium | Low"
},
"valence_morphology": {
"causative": "Yes | No | null",
"passive": "Yes | No | null",
"applicative": "Yes | No | null"
},
"numeral_classifiers": {
"value": "Classifier language | Non-classifier | null",
"confidence": "High | Medium | Low"
}
}
}
```

Figure 11: Diversity evaluation prompt.

```
You are a senior linguist specializing in constructed languages. Your task is to check the internal
consistency of a single translated sentence from a conlang, examining how well the translation aligns
with the given phonology, grammar, and lexicon.
1. Carefully read the three language sections (PHONOLOGY, GRAMMAR, LEXICON).
2. Examine the provided sentence translation, including the conlang sentence, gloss, and claimed
new words/grammar rules.
3. Identify every inconsistency, error, or mismatch you can find in this specific translation (for
example: phonotactic violations, grammatical rule violations, lexicon mismatches, gloss errors,
etc.).
Only account for linguistic errors, not formatting or stylistic errors.
4. For each issue, decide how serious it is:
• minor – cosmetic problems that don't affect comprehension or linguistic accuracy
• moderate – noticeable errors that affect clarity or violate established rules but don't break the
translation entirely
• major – serious violations that make the translation incorrect or incomprehensible given the
language system
5. Produce your answer ONLY in the JSON format specified below. No additional keys, no comments,
no markdown fences.
Required output JSON format:
{
"inconsistencies": [
{
"area": "string (Phonology | Grammar | Lexicon | Translation | Gloss | Cross-domain)",
"description": "string (describe what is inconsistent)",
"severity": "string (minor | moderate | major)"
}
. . . (0 or more items)
],
"final_verdict": "string (consistent | mostly consistent | inconsistent)"
}
Guidelines for the final verdict:
• If no inconsistencies are found, return "consistent".
• If only minor issues are found, or one moderate issue, return "mostly consistent".
• If any major issue, or multiple moderate issues are found, return "inconsistent".
=== LANGUAGE DATA STARTS ===
Phonology:
=== START ===
{phonology}
=== END ===
Grammar:
=== START ===
{grammar}
=== END ===
Lexicon:
=== START ===
{lexicon}
=== END ===
=== TRANSLATION TO EVALUATE ===
English sentence: {english_sentence}
Conlang translation: {conlang_sentence}
Gloss: {gloss}
New words claimed: {new_words}
New grammar rules claimed: {new_grammar_rules}
```

Figure 12: Consistency evaluation prompt.

Phonology

Consonants

	Labial	Alveolar	Palatal	Velar	Uvular	Glottal
Plosive	p	t		k	q	
Ejective	pʼ	tʼ		kʼ	qʼ	
Nasal	m	n				
Prenasalized Stop	ᵐb	ⁿd		ᵑg		
Fricative		s				h
Liquid		l				
Glide			j	w		

Vowels

	Front	Central	Back
High (+ATR)	i		u
High (-ATR)	ɪ		ʊ
Mid (+ATR)	e		o
Mid (-ATR)	ɛ		ɔ
Low (-ATR)		a	

Phonotactics

The language has a strict CV syllable structure. All words consist of one or more CV syllables, with no consonant clusters, codas, or vowel-initial words permitted. There are significant co-occurrence restrictions: ejectives (/pʼ, tʼ, kʼ, qʼ/) may only be followed by -ATR vowels (/ɪ, ʊ, ɛ, ɔ, a/), while prenasalized stops (/ᵐb, ⁿd, ᵑg/) may only be followed by +ATR vowels (/i, u, e, o/). The low vowel /a/ is neutral to front/back harmony but acts as a -ATR vowel for the purposes of consonant co-occurrence and ATR harmony.

Suprasegmentals

The language has no phonemic tone. Primary stress is fixed on the first syllable of a word. Two interlocking vowel harmony systems are present: front/back harmony and ATR/RTR harmony. All vowels in a word must agree in tongue root position (either all +ATR or all -ATR). Within a word, all vowels other than /a/ must also agree in backness (either all front or all back). The low vowel /a/ is exempt from the front/back harmony requirement and can co-occur with a series of front vowels or a series of back vowels, provided the ATR harmony is maintained. These harmony rules are triggered by the root's vowels and apply to all affixes.

Word Shapes and Lexical Statistics

All words in the language are composed of one or more CV syllables. The smallest possible word is a single CV syllable, and there is no theoretical maximum length, though words longer than four or five syllables are rare.

Monosyllabic and disyllabic words are very common and constitute the vast majority of the lexicon. Most function items are monosyllabic, while most content words are disyllabic or trisyllabic. Longer words are almost exclusively derived or compounded content words.

The strict vowel harmony systems create distinct "sets" of vowels that can appear in a given word: front +ATR, front -ATR, back +ATR, and back -ATR. The vowel /a/ can appear with either the front -ATR set or the back -ATR set. This, combined with the co-occurrence restrictions on ejective and prenasalized consonants, means that certain phoneme combinations are common while others are impossible. For instance, words containing prenasalized stops will only ever feature the vowels /i, u, e, o/, while words with ejectives will only ever feature /ɪ, ʊ, ɛ, ɔ, a/. This gives words containing these consonant types a distinct phonetic character. Words with only plain plosives, nasals, fricatives, liquids, and glides are the most numerous and phonologically flexible.

Lexical Items

Content Items (Common)

* /ˈkʼɔsa/
* /ˈⁿdoku/
* /ˈpɛlɛ/
* /ˈhuti/
* /ˈqʼala/
* /ˈmeⁿdile/
* /ˈsɔtʼa/
* /ˈwɛjɛ/
* /ˈᵐbulo/
* /ˈjɔkʼɔ/
* /ˈquno/
* /ˈtʼʊha/
* /ˈpʼɛla/

Content Items (Uncommon)

* /ˈpʼɔtʼa/
* /ˈᵑgoᵐbiso/
* /ˈtʼɛsɪlɪ/
* /ˈqʼʊkʼatʼa/
* /ˈwuwolo/
* /ˈjɪjɛhɛ/

Function Items (Common)

* /ˈse/
* /ˈkɔ/
* /ˈna/
* /ˈtu/
* /ˈlɛ/

Figure 13: Sample language sketch, page 1/7.

Function Items (Uncommon)

* /ˈqˈɔ/

* /ˈⁿdo/

Grammar

Morphology

The language is primarily analytic, with most grammatical information conveyed by word order and free-standing particles. Nouns are invariant and do not decline for case or number. The key exception to the analytic structure is the verb, which is strictly head-marking and carries a small set of agglutinative prefixes that cross-reference its core arguments.

The language has an active-stative morphosyntactic alignment. Verbs are obligatorily prefixed to indicate the semantic role of their arguments. There are two sets of prefixes: an Agentive set (for volitional actors) and a Patientive set (for entities that are in a state or are affected by an action). The choice of prefix for an intransitive verb depends on the semantics of the verb and its subject.

* An intransitive verb with a volitional agent takes an Agentive prefix (`Sa`).

* An intransitive verb with a patient-like subject takes a Patientive prefix (`Sp`).

* A transitive verb takes both an Agentive (`A`) and a Patientive (`P`) prefix, in that order.

The prefixes are single CV syllables and obey the language's vowel harmony rules, assimilating to the front/back and ATR quality of the verb root.

| Role | Prefix Form | Example (with root `hoto` "run") | Example (with root `kˈɔsa` "break") |

| :--- | :--- | :--- | :--- |

| **Agentive** | `to-` / `tʊ-` | `to-hoto` | `tʊ-kˈɔsa` |

| **Patientive** | `no-` / `nɔ-` | `no-hoto` | `nɔ-kˈɔsa` |

To-hoto hoto.

```
to-hoto      hoto
AGT-run      man
"The man runs." (volitional)
```

Nɔ-kˈɔsa pˈɛla.

```
nɔ-kˈɔsa     pˈɛla
PAT-break    rock
"The rock broke." (stative)
```

Tʊ-nɔ-kˈɔsa hoto pˈɛla.

```
tʊ-nɔ-kˈɔsa     hoto  pˈɛla
AGT-PAT-break   man   rock
"The man broke the rock."
```

Tense, aspect, and mood are handled by pre-verbal particles. The system is minimal, distinguishing only Realis (the default, unmarked state) from Irrealis, which covers future, hypothetical, and potential events. The Irrealis particle is `se`.

Se to-hoto hoto.

```
se   to-hoto      hoto
IRR  AGT-run      man
"The man will run / might run."
```

Verbal Derivation

The verb root can be modified by derivational suffixes to change its meaning. These suffixes, like the prefixes, obey the language's vowel harmony rules. They are added directly to the root.

1. Causative Suffix `-lo` / `-lɔ`

The causative suffix adds an agent that causes the action to occur, increasing the verb's valency. The newly introduced causer is always marked by the Agentive prefix (`to-`/`tʊ-`), which harmonizes with the verb root as usual.

* **Intransitive to Transitive:** When added to an intransitive verb, it makes it transitive.
  * *Nɔ-kˈɔsa pˈɛla.* ("The rock broke.")
  * *Tʊ-nɔ-kˈɔsa-lɔ hoto pˈɛla.*

    ```
    tʊ-nɔ-kˈɔsa-lɔ     hoto  pˈɛla
    AGT-PAT-break-CAUS  man   rock
    "The man made the rock break."
    ```

    (This is semantically distinct from the base transitive form, emphasizing the causing of a state rather than a direct action.)

* **Transitive to Ditransitive:** When added to a transitive verb, it makes it ditransitive. The original agent is demoted to a secondary object (the "causee"), and the original patient remains the patient. The original Agentive prefix is replaced by one marking the new causer, while the Patientive prefix is retained to mark the original patient.

Figure 14: Sample language sketch, page 2/7.

*   *Tʊ-nɔ-sɔtʼa wɛjɛ jɔkʼɔ.* ("The child saw the dog.")

*   *Tʊ-nɔ-sɔtʼa-lɔ hoto wɛjɛ jɔkʼɔ.*

    ```

    tʊ-nɔ-sɔtʼa-lɔ     hoto  wɛjɛ  jɔkʼɔ

    AGT-PAT-see-CAUS     man   child dog

    "The man made the child see the dog." / "The man showed the dog to the child."

    ```

2. Reciprocal Suffix `-so` / `-sɔ`

The reciprocal suffix indicates that the action is performed by members of a group on each other. It is used with a plural or coordinated subject and requires an Agentive prefix. Because the action is directed at another agent within the subject group, the verb becomes syntactically intransitive.

*   *Tʊ-sɔtʼa-sɔ hoto na wɛjɛ.*

    ```

    tʊ-sɔtʼa-sɔ       hoto na wɛjɛ

    AGT-see-RECIP      man and child

    "The man and the child see each other."

    ```

3. Stative/Adjectival Suffix `-mo`/`-mɔ` and `-me`/`-mɛ`

This suffix derives a stative adjective from a verb root. The resulting word can be used as a modifier in a noun phrase. The suffix has four forms to obey vowel harmony: `-mo` (back, +ATR), `-mɔ` (back, -ATR), `-me` (front, +ATR), `-mɛ` (front, -ATR).

*   **Verb root:** `kʼɔsa` ("to break")

*   **Stative Adjective:** `kʼɔsamɔ` ("broken")

    Tʊ-nɔ-sɔtʼa wɛjɛ quno kʼɔsamɔ.

    ```

    tʊ-nɔ-sɔtʼa     wɛjɛ  quno kʼɔsamɔ

    AGT-PAT-see      child house broken

    "The child sees the broken house."

    ```

    (The derived adjective `kʼɔsamɔ` contains an ejective from its root, so it follows the noun it modifies, obeying the phonologically conditioned syntax rule.)

Pronouns and Pro-Drop

Because the verb's prefixes clearly mark the agent and patient, overt noun phrases for the subject and object are often dropped when understood from context (pro-drop). Independent pronouns exist for emphasis, contrast, or clarity. They are not marked for case; their role is understood from the verb's prefixes. There is no gender distinction.

| | Singular | Plural |

| :--- | :---: | :---: |

| **1st Person** | `me` | `meje` |

| **2nd Person** | `so` | `soje` |

| **3rd Person** | `hɛ` | `hɛje` |

*   *To-hoto.* ("He/she runs.")

*   *To-hoto **hɛ**.*

    ```

    to-hoto      hɛ

    AGT-run      3SG

    "**He/She** runs." (in contrast to someone else)

    ```

*   *Tʊ-nɔ-sɔtʼa **me** **hɛ**.*

    ```

    tʊ-nɔ-sɔtʼa     me  hɛ

    AGT-PAT-see      1SG 3SG

    "**I** see **him/her**."

    ```

Negation

Negation is marked by a particle, `kʼa`, which is placed directly after the verb complex (i.e., after the prefixed verb and any derivational suffixes). The presence of `kʼa` forces the entire verb complex it modifies (both prefixes and root) to take on [-ATR] vowel harmony.

*   **Affirmative:** *To-hoto hoto.* ("The man runs.")

    (The verb `to-hoto` is [+ATR].)

*   **Negative:** *Tʊ-hɔtɔ kʼa hoto.*

    ```

    tʊ-hɔtɔ      kʼa  hoto

    AGT-run      NEG  man

    "The man does not run."

    ```

    (The negative particle `kʼa` forces the entire verb `to-hoto` to become its [-ATR] counterpart, `tʊ-hɔtɔ`.)

Syntax

The basic constituent order is Verb-Subject-Object (VSO). Overt noun phrases for the subject and object are often dropped when understood from context.

Figure 15: Sample language sketch, page 3/7.

Tʊ-nɔ-sɔtʼa wɛjɛ jɔkʼɔ.

```
tʊ-nɔ-sɔtʼa     wɛjɛ  jɔkʼɔ
AGT-PAT-see     child dog
"The child sees the dog."
```

Tʊ-nɔ-sɔtʼa.

```
tʊ-nɔ-sɔtʼa
AGT-PAT-see
"(He/she/it) sees (him/her/it)."
```

The language uses postpositions almost exclusively. A postposition forms a phrase with the preceding noun.

To-hoto hoto quno kɔ.

```
to-hoto      hoto  quno kɔ
AGT-run      man   house in
"The man runs in the house."
```

Noun Phrase Structure and Ejective Displacement

Noun phrase structure is typically head-final: adjectives and genitives (possessors) precede the noun they modify (Modifier-Possessor-Noun).

Tʊ-nɔ-sɔtʼa ᵐbulo wɛjɛ pɛlɛ jɔkʼɔ.

```
tʊ-nɔ-sɔtʼa     ᵐbulo wɛjɛ    pɛlɛ  jɔkʼɔ
AGT-PAT-see     big   child   small dog
"The big child sees the small dog."
```

However, this order is subject to a phonologically conditioned rule called **Ejective Displacement**. Any item within a noun phrase that contains an ejective consonant—be it a modifier, possessor, or the head noun itself—disrupts the standard head-final NP order.

1. **Standard Order (Modifier-Possessor-Noun):**
   * *ᵐbulo hoto quno*

   ```
   ᵐbulo hoto  quno
   big   man   house
   "the big man's house"
   ```

2. **Ejective in Modifier/Possessor (Displaces Element):** A modifier or possessor containing an ejective must follow the rest of the noun phrase.
   * *hoto quno pʼɛla*

   ```
   hoto  quno pʼɛla
   man   house small
   "the man's small house" (adjective `pʼɛla` has /pʼ/)
   ```

   * *quno ᵐbulo tʼʊha*

   ```
   quno ᵐbulo tʼʊha
   house big  chief
   "the chief's big house" (possessor `tʼʊha` "chief" has /tʼ/)
   ```

   * *Tʊ-nɔ-sɔtʼa wɛjɛ jɔkʼɔ qʼala.*

   ```
   tʊ-nɔ-sɔtʼa     wɛjɛ  jɔkʼɔ qʼala
   AGT-PAT-see     child dog   bad
   "The child sees the bad dog." (Adjective `qʼala` follows noun due to `/qʼ/`.)
   ```

3. **Ejective in Head Noun (Inverts Entire Phrase):** If the head noun itself contains an ejective, the entire phrase inverts to a head-initial structure. The head noun comes first, followed by its possessors and then its other modifiers.
   * **Noun:** `jɔkʼɔ` ("dog", no ejective) -> *pɛlɛ jɔkʼɔ* ("small dog")
   * **Noun:** `pʼɛla` ("rock", has /pʼ/) -> *pʼɛla ᵐbulo* ("the big rock")
   * **Complex Example:** `pʼɛla hoto ᵐbulo`

   ```
   pʼɛla  hoto  ᵐbulo
   rock   man   big
   "the man's big rock"
   ```

Question Formation

1. Yes/No Questions

Yes/no questions are formed with the sentence-final particle `he`. This particle is phonologically neutral and does not affect harmony.

Figure 16: Sample language sketch, page 4/7.

*   *Tʊ-nɔ-sɔtʼa wɛjɛ jɔkʼɔ **he**?*
    ```
    tʊ-nɔ-sɔtʼa  wɛjɛ  jɔkʼɔ he
    AGT-PAT-see  child dog  Q
    "Does the child see the dog?"
    ```

2. Content (Wh-) Questions

Content questions use question words that are fronted to the beginning of the sentence, before the verb. This is an exception to the standard VSO word order.

*   **Common Question Words:**
    *   `hajeˋ ("who?", "what?" - for animates)
    *   `namaˋ ("what?" - for inanimates)
    *   `lɛleˋ ("where?")
    *   `seseˋ ("when?")
    *   `kokeˋ ("why?")

*   ***Haje** tʊ-nɔ-sɔtʼa jɔkʼɔ?*
    ```
    haje  tʊ-nɔ-sɔtʼa  jɔkʼɔ
    who  AGT-PAT-see  dog
    "Who sees the dog?"
    ```

*   ***Nama** tʊ-nɔ-sɔtʼa wɛjɛ?*
    ```
    nama  tʊ-nɔ-sɔtʼa  wɛjɛ
    what  AGT-PAT-see  child
    "What does the child see?"
    ```

*   ***Lɛle** to-hoto hoto?*
    ```
    lɛle  to-hoto  hoto
    where AGT-run  man
    "Where does the man run?"
    ```

Subordination and Clause Chaining

Subordination is handled through clause chaining and medial verbs. The bare verb root serves as a medial verb in two distinct constructions.

1. Medial Verbs in Clause Chains

A series of clauses can be linked together, with only the final verb in the chain being fully prefixed. All preceding, non-final verbs are "medial verbs," which appear as bare roots. These medial verbs share their subject with the following clause.

hoto, sɔtʼa pʼɛla, to-no-quno.
```
run,  see  rock,  AGT-PAT-take
"(He) ran, saw the rock, and took it."
```

2. Medial Verbs in Relative Clauses

Relative clauses are formed without a relative pronoun. The modifying clause is placed directly before the noun it modifies, and its verb is in the bare root (medial verb) form. The medial verb shares its agent with the head noun it modifies.

*   *Tʊ-nɔ-sɔtʼa wɛjɛ [**sɔtʼa jɔkʼɔ**] hoto.*
    ```
    tʊ-nɔ-sɔtʼa  wɛjɛ  [sɔtʼa jɔkʼɔ] hoto
    AGT-PAT-see  child [see  dog]  man
    "The child sees the man who saw the dog."
    ```

word,translation,pos,notes

-lo/-lɔ,suffix,"Causative suffix",Verbal derivational suffix. Harmonizes for ATR and backness with the verb root.,,-lo (+ATR back), -lɔ (-ATR back), -le (+ATR front), -lɛ (-ATR front)

-mo/-mɔ/-me/-mɛ,suffix,"Stative/Adjectival suffix",Verbal derivational suffix. Harmonizes for ATR and backness with the verb root.,,kʼɔsa (v) > kʼɔsamɔ (adj),-mo (+ATR back), -mɔ (-ATR back), -me (+ATR front), -mɛ (-ATR front)

-so/-sɔ/-se/-sɛ,suffix,"Reciprocal suffix",Verbal derivational suffix. Harmonizes for ATR and backness with the verb root.,,-so (+ATR back), -sɔ (-ATR back), -se (+ATR front), -sɛ (-ATR front)

/ˈha/,postposition,"with; using",[-ATR] neutral vowel.,,

/ˈhopo-lo/,verb,"send; make go",Requires [+ATR] back harmony verb root.,Derived from verb root 'hopo' + causative suffix '-lo'.,

/ˈhopo-so/,verb,"go together; travel together",Requires [+ATR] back harmony verb root.,Derived from verb root 'hopo' + reciprocal suffix '-so'.,

/ˈhopo/,verb,"go",Requires [+ATR] back harmony verb root.,,

/ˈhɑnɛ/,verb,"laugh",Requires [-ATR] front harmony verb root.,,

/ˈjɛpɛ/,noun,"hand",Requires [-ATR] front harmony.,,

/ˈkɛla/,noun,"tree",Requires [-ATR] front harmony.,,

/ˈkite/,noun,"knife",Requires [+ATR] front harmony.,,

/ˈkʼatʼa/,noun,"fire",Requires [-ATR] back harmony. Contains ejectives /kʼ/ and /tʼ/, triggering Ejective Displacement when it is the head noun.,,

Figure 17: Sample language sketch, page 5/7.

/ˈkʾaⁿda/,verb,"sleep",Requires [-ATR] back harmony verb root. Contains an ejective /kʾ/ and a prenasalized stop /ⁿd/.,,Irregular: Contains a prenasalized stop /ⁿd/ followed by a [-ATR] vowel /a/, violating the rule that prenasalized stops must be followed by [+ATR] vowels.

/ˈkʾi/,particle,"completive aspect marker",Post-verbal particle. Forces the verb complex to take [-ATR] harmony. Contains ejective /kʾ/.,,This particle forces [-ATR] harmony. Example: to-pete > tʊ-petɛ kʾi.

/ˈkʾɔɔ/,adjective,"hot",Requires [-ATR] back harmony. Contains ejective /kʾ/, triggering Ejective Displacement.,,

/ˈkʾɛlɛ-sɛ/,verb,"sing together",Requires [-ATR] front harmony verb root. Contains ejective /kʾ/.,Derived from verb root 'kʾɛlɛ' (to sing) + reciprocal suffix '-sɛ'.,

/ˈkʾɛlɛ/,verb,"sing",Requires [-ATR] front harmony verb root. Contains ejective /kʾ/.,,

/ˈkʾʊlɔ/,verb,"drink",Requires [-ATR] back harmony verb root. Contains ejective /kʾ/.,,

/ˈlolo/,adjective,"red",Requires [+ATR] back harmony.,,

/ˈlɔkʾo/,noun,"blood",Requires [-ATR] back harmony. Contains ejective /kʾ/, triggering Ejective Displacement when it is the head noun.,,

/ˈlɛpɛ-lɛ/,verb,"feed",Requires [-ATR] front harmony verb root.,Derived from verb root 'lɛpɛ' + causative suffix '-lɛ'.,

/ˈlɛpɛ/,verb,"eat",Requires [-ATR] front harmony verb root.,,

/ˈmuna/,noun,"moon; month",Requires [+ATR] back harmony.,,

/ˈmɛkɛ/,noun,"eye",Requires [-ATR] front harmony.,,

/ˈnɛ/,postposition,"on",[-ATR] front vowel.,,

/ˈpete-le/,verb,"catch (a fish); make swim",Requires [+ATR] front harmony verb root.,Derived from verb root 'pete' (to swim) + causative suffix '-le'.,The verb root 'pete' is inferred to be polysemous with the noun 'pete' (fish).

/ˈpete/,noun,"fish",Requires [+ATR] front harmony.,,

/ˈpete/,verb,"swim",Requires [+ATR] front harmony verb root.,,Inferred from the noun 'pete' (fish).

/ˈpete/,verb,"speak",Requires [+ATR] front harmony verb root.,,

/ˈpʾajɛ/,verb,"give",Requires [-ATR] front harmony verb root. Contains ejective /pʾ/.,,

/ˈpʾatʾa/,noun,"head",Requires [-ATR] back harmony. Contains ejectives /pʾ/ and /tʾ/, triggering Ejective Displacement when it is the head noun.,,

/ˈpʾʊsʊ/,noun,"mouth",Requires [-ATR] back harmony. Contains ejective /pʾ/, triggering Ejective Displacement when it is the head noun.,,

/ˈqʾatʾa-lɔ/,verb,"burn; set on fire",Requires [-ATR] back harmony verb root. Contains ejectives /qʾ/ and /tʾ/.,Derived from verb root 'qʾatʾa' (to be on fire) + causative suffix '-lɔ'.,

/ˈqʾatʾa/,verb,"be on fire",Requires [-ATR] back harmony verb root. Contains ejectives /qʾ/ and /tʾ/.,,Inferred from the noun 'kʾatʾa' (fire), with a plausible phonological shift from /kʾ/ to /qʾ/ to distinguish the verb.

/ˈqʾɛsɛ/,adjective,"white",Requires [-ATR] front harmony. Contains ejective /qʾ/, triggering Ejective Displacement.,,

/ˈsani/,adjective,"good",Requires [+ATR] front harmony.,,

/ˈsɔla/,noun,"sun; day",Requires [-ATR] back harmony.,,

/ˈsɔtʾamɔ/,adjective,"visible; seen",Requires [-ATR] back harmony. Contains ejective /tʾ/ from its root, triggering Ejective Displacement.,Derived from verb root 'sɔtʾa' + stative suffix '-mɔ'.,

/ˈsɛkʾɛ/,noun,"leaf",Requires [-ATR] front harmony. Contains ejective /kʾ/, triggering Ejective Displacement when it is the head noun.,,

/ˈteli/,adjective,"cold",Requires [+ATR] front harmony.,,

/ˈtʾaka-mɔ/,adjective,"pierced; speared",Requires [-ATR] back harmony. Contains ejective /tʾ/ from its root, triggering Ejective Displacement.,Derived from verb root 'tʾaka' (to spear) + stative suffix '-mɔ'.,

/ˈtʾaka/,noun,"spear",Requires [-ATR] back harmony. Contains ejective /tʾ/, triggering Ejective Displacement when it is the head noun.,,

/ˈtʾaka/,verb,"spear; pierce",Requires [-ATR] back harmony verb root. Contains ejective /tʾ/.,,Inferred from the noun 'tʾaka' (spear).

/ˈtʾɛsɛlɛmɛ/,adjective,"cut; sharp",Requires [-ATR] front harmony. Contains ejective /tʾ/ from its root, triggering Ejective Displacement.,Derived from verb root 'tʾɛsɛ' (to cut) + causative suffix '-lɛ' + stative suffix '-mɛ'.,

/ˈtʾɛlɛ/,noun,"foot",Requires [-ATR] front harmony. Contains ejective /tʾ/, triggering Ejective Displacement when it is the head noun.,,

/ˈtʾɪpʾa/,noun,"stone",Requires [-ATR] front harmony. Contains ejectives /tʾ/ and /pʾ/, triggering Ejective Displacement when it is the head noun.,,

/ˈwɔpʾa/,noun,"sky",Requires [-ATR] back harmony. Contains a historical ejective /pʾ/, and still triggers Ejective Displacement as an irregularity.,,Irregular: Triggers Ejective Displacement despite lacking an overt ejective.

/ˈwɔta/,noun,"cloud",Requires [-ATR] back harmony.,,

/ˈwɔla/,verb,"die",Requires [-ATR] back harmony verb root.,,

/ˈwɔlamɔ/,adjective,"dead",Requires [-ATR] back harmony.,Derived from verb root 'wɔla' + stative suffix '-mɔ'.,

/ˈᵐbeke/,noun,"song",Requires [+ATR] front harmony. Contains prenasalized stop /ᵐb/.,,

/ˈᵐbubu/,adjective,"black",Requires [+ATR] back harmony. Contains prenasalized stop /ᵐb/.,,

/ˈᵑgaja/,noun,"river",Requires [-ATR] back harmony. Contains a prenasalized stop /ᵑg/.,,Irregular: Contains a prenasalized stop but has [-ATR] vowels.

/ˈᵑginɔ/,verb,"think",Requires [+ATR] front harmony verb root. Contains prenasalized stop /ᵑg/.,,

/ˈᵑgene/,noun,"path; road",Requires [+ATR] front harmony. Contains prenasalized stop /ᵑg/.,,

/ˈⁿdemi/,verb,"know",Requires [+ATR] front harmony verb root. Contains prenasalized stop /ⁿd/.,,

/ˈⁿdele/,adjective,"long",Requires [+ATR] front harmony. Contains prenasalized stop /ⁿd/.,,

/ˈⁿdumɔ/,noun,"heart",Requires [+ATR] back harmony. Contains prenasalized stop /ⁿd/.,,

haje,interrogative pronoun,"who; what (animate)",Sentence-initial question word.,,

he,particle,"yes/no question marker",Sentence-final particle.,,

hoto,noun,"man",Requires [+ATR] back harmony.,,

huto,noun,"woman",Requires [+ATR] back harmony.,,

hɛ,pronoun,"he; she; it (3rd person singular)",,

hɛje,pronoun,"they; them (3rd person plural)",Derived from 'hɛ' + pluralizer '-je'.,

jɔkʾɔ,noun,"dog",Requires [-ATR] back harmony. Contains ejective /kʾ/, triggering Ejective Displacement when it is the head noun.,,

jɪjɛhɛ,noun,"bird",Requires [-ATR] front harmony.,,

koke,interrogative adverb,"why",Sentence-initial question word.,,

kɔ,postposition,"in; at",[-ATR] back vowel.,,

kʾa,particle,"negation marker",Post-verbal particle. Forces the verb complex to take [-ATR] harmony. Contains ejective /kʾ/.,,

kʾɔsa,verb,"break",Stative intransitive verb root; requires [-ATR] back harmony. Contains ejective /kʾ/.,,

kʾɔsamɔ,adjective,"broken",Requires [-ATR] back harmony. Contains ejective /kʾ/ from its root, triggering Ejective Displacement.,Derived from verb root 'kʾɔsa' + stative suffix '-mɔ'.,

Figure 18: Sample language sketch, page 6/7.

lɛ,postposition,"from",[-ATR] front vowel.,,

lɛle,interrogative adverb,"where",Sentence-initial question word.,,

me,pronoun,"I; me (1st person singular)",,

meje,pronoun,"we; us (1st person plural)",Derived from 'me' + pluralizer '-je'.,

meⁿdile,noun,"food",Requires [+ATR] front harmony. Contains prenasalized stop /ⁿd/.,,

na,conjunction,"and",[-ATR] neutral vowel.,,

nama,interrogative pronoun,"what (inanimate)",Sentence-initial question word.,,

no-/nɔ-,prefix,"Patientive prefix (Sp/P)",Verbal prefix for patients/stative subjects. Harmonizes for ATR and backness with the verb root.,,no- (+ATR), nɔ- (-ATR)

/ˈpʾɛla/,noun,"rock",Requires [-ATR] front harmony. Contains ejective /pʾ/, triggering Ejective Displacement when it is the head noun.,,

/ˈpʾɛla/,adjective,"small",Requires [-ATR] front harmony. Contains ejective /pʾ/, triggering Ejective Displacement.,,

pʾɔtʾa,verb,"hit",[-ATR] back harmony verb root. Contains ejective /pʾ/ and /tʾ/.,,

quno,noun,"house",Requires [+ATR] back harmony.,,

quno,verb,"take",[+ATR] back harmony verb root.,,Example of use: to-no-quno. Gloss: AGT-PAT-take. "(He/she) took (it)".

qʾala,adjective,"bad",Requires [-ATR] back harmony. Contains ejective /qʾ/, triggering Ejective Displacement.,,

qʾɔ,particle,"emphatic marker",[-ATR] back vowel. Contains ejective /qʾ/.,,

qʾukʾatʾa,verb,"crush",[-ATR] back harmony verb root. Contains ejectives /qʾ/, /kʾ/, /tʾ/.,,

se,particle,"irrealis marker",Pre-verbal particle marking future, hypothetical, or potential events. [+ATR] front vowel.,,

sese,interrogative adverb,"when",Sentence-initial question word.,,

so,pronoun,"you (2nd person singular)",,

soje,pronoun,"you (2nd person plural)",Derived from 'so' + pluralizer '-je'.,

sɔtʾa,verb,"see",[-ATR] back harmony verb root. Contains ejective /tʾ/.,,

to-/tu-,prefix,"Agentive prefix (Sa/A)",Verbal prefix for volitional agents. Harmonizes for ATR and backness with the verb root.,,to- (+ATR), tʊ- (-ATR)

tu,postposition,"to; for",Requires [+ATR] back harmony.,,

tʾɛsɛ,verb,"cut",[-ATR] front harmony verb root. Contains ejective /tʾ/.,,

tʾʊha,noun,"chief",Requires [-ATR] back harmony. Contains ejective /tʾ/, triggering Ejective Displacement when it is the head noun.,,

wuwolo,noun,"water",Requires [+ATR] back harmony.,,

wɛjɛ,noun,"child",Requires [-ATR] front harmony.,,

ᵐbulo,adjective,"big",Requires [+ATR] back harmony. Contains prenasalized stop /ᵐb/.,,

ᵑgoᵐbiso,verb,"gather",Requires [+ATR] back harmony. Contains prenasalized stops /ᵑg/ and /ᵐb/.,,

ⁿdo,particle,"topic marker",Requires [+ATR] back harmony. Contains prenasalized stop /ⁿd/.,,

ⁿdoku,noun,"stick",Requires [+ATR] back harmony.,,

Translation

English sentence: "The big dog is sleeping."

Conlang sentence: "nɔ-kʾaⁿda jɔkʾɔ. to-hoto ᵐbulo hɛ."

Gloss: "PAT-sleep dog.SG.ABS 3SG.N.ABS-be.STAT big.ADJ PRF"

New words:

"to-": "3rd person singular non-human absolutive prefix"

"hoto": "stative verb 'to be'"

"hɛ": "perfective aspect particle"

New grammar rules:

Rule: "Stative Predication for Adjectives"

Justification: "When an adjective cannot agree in ATR harmony with the noun it modifies, it cannot form a direct noun phrase. Instead, the adjective is predicated of the noun in a separate stative clause using the verb 'hoto' and a resumptive pronoun. This resolves the harmony clash while preserving the meaning, a common strategy in languages with strong vowel harmony."

English sentence: "Where is my book?"

Conlang sentence: "Lɛle nɔ-kɔpɔ sɔpɔ me"

Gloss: " where PAT-exist book-1SG"

New_words:

"kɔpɔ": "to be (in a place), exist""hoto": "stative verb 'to be'"

"sɔpɔ": "book"

New grammar rules:

Rule: "Possession is indicated by juxtaposing a pronoun after the noun it possesses (Noun-Pronoun)."

Justification: "The language specification provides a Possessor-Noun structure for nominal possessors (e.g., 'hoto quno' for 'the man's house'), but does not specify how pronominal possession is handled. This new rule establishes a distinct structure for pronouns, which often follow different syntactic rules than full noun phrases. Placing the pronoun post-nominally is a common typological pattern that avoids ambiguity. This Noun-Pronoun order takes precedence over the Ejective Displacement rule, meaning the pronoun always follows the noun it possesses, regardless of ejectives in the noun."

Figure 19: Sample language sketch, page 7/7.