AvarLab: An Integrated Digital Ecosystem for Avar, a Morphologically Rich Low-Resource Language

Kebed Zagidov
Universitat Pompeu Fabra
Barcelona, Spain
kebed.zagidov@upf.edu

Thomas Brochhagen
Universitat Pompeu Fabra
Barcelona, Spain
thomas.brochhagen@upf.edu

Abstract

Many low-resource languages remain digitally under-resourced not only because of limited data, but also because lexical resources, corpora, and computational tools are typically developed in isolation. We present AvarLab, an integrated platform for Avar, a morphologically rich Northeast Caucasian language. The system implements a generate–verify workflow: lexical entries are expanded into inflectional paradigms, automatically annotated with grammatical features, and then verified against both corpus attestations and based on community-driven feedback. This approach supports dictionary lookup across inflected forms, corpus-based example retrieval, and the creation of silver-standard morphological annotations for downstream NLP applications. In its current version, AvarLab covers 14,768 lexical entries and generates over a million inflected forms across parts of speech. We argue that tightly integrating lexicography, corpus verification, and community validation provides a practical pathway for building computational infrastructure for morphologically rich low-resource languages.

1 Introduction

In many multilingual contexts, when one language comes to dominate public life, others retreat to family and local community contexts. This can lead to the marginalization of such minority languages, particularly in digital environments. As communication, education, and knowledge exchange increasingly move online, languages that lack digital infrastructure risk further marginalization (Kornai, 2013).

This challenge becomes evident for an Avar speaker attempting to engage with their language online today. Resources are scarce, and digital platforms offer little or no support for the language. Even basic digital communication requires improvisation: users routinely rely on nonstandard spellings to represent sounds or letters missing from standard keyboards. As a result, speakers, especially younger generations, frequently shift toward languages that are easier to use in digital spaces. The situation is also challenging when trying to build NLP resources for Avar. Although formal aspects of Avar are described in the linguistic literature, the resources necessary to train modern NLP tools remain fragmented. Morphological descriptions are scattered across descriptive grammars, annotated corpora are extremely limited, and available lexical resources are largely static digitizations of printed dictionaries (Alekseev et al., 2012; Forker, 2017; Alikhanov, 2003). Without structured datasets and computational models, Avar remains almost entirely absent from contemporary NLP pipelines.

Avar, a Northeast Caucasian language spoken primarily in the Republic of Dagestan, Russia, and classified by UNESCO as vulnerable (Moseley, 2010), illustrates the challenge faced by many morphologically rich, low-resource languages: While NLP leaps forward, languages such as Avar remain marginalized within the digital ecosystem (Joshi et al., 2020). This challenge is not only due to lack of data but also due to the linguistic complexity of the language itself. Avar exhibits ergative–absolutive alignment, extensive nominal case marking, and a pervasive class-based agreement system across four grammatical classes (Class I: masculine, Class II: feminine, Class III: objects/animals, and Plural) (Alekseev et al., 2012; Forker, 2017). These characteristics are further complicated by unpredictable oblique stem alternations and a highly productive spatial case system that can generate dozens of distinct forms for a single noun (Alekseev et al., 2012; Forker, 2017; Khangereev, 2011). While these features make the language typologically rich and expressive, they also produce extreme data sparsity for conventional NLP approaches.

Proceedings of the Ninth Workshop on the Use of Computational Methods in the Study of Endangered Languages (ComputEL-9), pages 62–71

The motivation for this work emerges from a dual perspective. As both a native speaker of Avar and a computational linguist, the first author encountered the limitations of existing digital resources first hand. What began as an effort to build a practical community-editable online dictionary soon revealed a systemic obstacle common to morphologically rich low-resource languages: a circular dependency. Large annotated corpora are required to train NLP models, yet such corpora cannot be created without pre-existing linguistic tools. (Magueresse et al., 2020; Nekoto et al., 2020). This led to the development of AvarLab[1], an integrated digital ecosystem designed to connect phonology, lexicography, morphological modeling, and corpus linguistics within a unified platform. At the core of the system is a generate–verify framework in which rule-based morphological models generate possible word forms, which are then verified against a growing corpus and refined through both automated analysis and community feedback. By transforming static descriptive grammar into an active computational pipeline, AvarLab bridges the gap between language documentation and modern NLP infrastructure.

The contributions of this paper are threefold. First, we introduce AvarLab, the first integrated digital ecosystem for Avar, combining trilingual lexicon, corpus resources, and rule-based morphological modeling. Second, we formalize a generate–verify workflow in which rule-based paradigm generation is coupled with corpus attestation and community validation. Third, we demonstrate how integrating lexicographic data, corpus evidence, and computational morphology can provide scalable infrastructure for developing NLP resources for morphologically rich low-resource languages.

2 Related work

While digital lexicography has advanced (Atkins and Rundell, 2008), platforms for Caucasian languages still largely reproduce printed materials. Online resources like `http://Avar.me` provide valuable lexical data but lack morphological coverage and corpus integration. Even Google Translate's recent addition of Avar relies heavily on Russian –a phylogenetically unrelated language– as a pivot, often failing on complex syntax due to the absence of robust structural modeling.

In morphological analysis, finite-state frameworks like HFST (Lindén et al., 2013) and comprehensive infrastructures built upon them, such as Giellatekno (Moshagen et al., 2014), perform well but require fully specified, static morphological metadata prior to compilation. In Avar, morpheme selection, for example, such as plural formation, is frequently conditioned by phonetics, semantics, or etymology rather than pure structural shape (Section 4.3.1). AvarLab addresses this limitation by using a rule-driven architecture that actively utilizes all available and computationally inferred metadata to drive generation. By encoding morphophonological rules directly, this approach enables immediate paradigm generation, while providing a foundation for future finite-state or neural-network implementations as resources grow.

From a corpus perspective, frequency-based validation is central to modern lexicography (Sinclair, 1991; Davies, 2008). However, Avar's largest dataset, AvarCorpora (Volina, 2023), remains limited in stylistic diversity and morphological annotation. Consequently, hybrid approaches combining automated generation with corpus-based validation are essential.

Although community-driven platforms like Language Hotspots (Anderson, 2011; Living Tongues Institute for Endangered Languages, n.d.), FirstVoices (First Peoples' Cultural Council, 2018), and Wikipedia (Giles, 2005) enable native speaker contributions, they typically operate without underlying computational morphology. AvarLab addresses this gap by interlinking rule-based generation, corpus annotation, and community validation within a unified workflow.

3 The Generate–Verify Framework

Morphologically rich languages present a major challenge for computational modeling. Extensive inflection generates large numbers of surface forms, yet such languages often lack the digital data needed to detect and represent them adequately. This creates a structural asymmetry: the linguistic system produces many forms, while available corpora contain only a small subset.

To address this problem, we propose a generate–verify framework that reverses the typical order of resource development. Instead of relying primarily on corpora to derive linguistic patterns, the system first generates complete morphological paradigms using rule-based models derived from descriptive grammars (Alekseev et al.,

[1] See `https://avardict.upf.edu`.

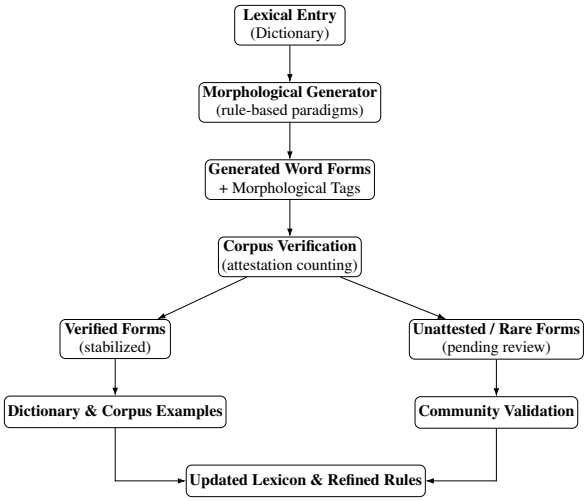

Figure 1: The generate–verify workflow implemented in AvarLab. Lexical entries are expanded into rule-based paradigms, automatically annotated, and verified against corpus evidence. Frequently attested forms are stabilized and linked to corpus examples, while unattested or rare forms remain open to community validation. Both pathways contribute to updating the lexicon and refining the morphological rules.

2012; Forker, 2017). These forms are then verified against corpus evidence and refined through iterative validation.

3.1 Conceptual Overview

As illustrated in Fig. 1, our framework operates as an iterative loop linking generation, annotation, verification, and community feedback. 1. Generation: Morphological rules derived from descriptive grammars generate complete paradigms for each lexical entry.

2. Annotation: Generated forms are automatically labeled for features such as part of speech, number, or class, forming a structured morphological database.

3. Verification: Generated forms are cross-checked against corpora to determine attestation.

4. Stabilization (Locking): Forms that meet the attestation threshold are "locked", preventing algorithmic overwriting or accidental modification.

5. Community feedback: Unattested or ambiguous forms are validated by native speakers through participatory interfaces.

In this way, generated forms are continuously evaluated and refined. As corpus coverage grows and speaker contributions accumulate, the system gradually converges toward increasingly accurate morphological representations.

3.2 Implementation in AvarLab

AvarLab's backend is built with Django and PostgreSQL. The platform integrates a lexical database, a rule-based morphological generator, and a corpus verification module within a relational architecture.

Each lexical entry serves as the starting point for paradigm generation. Based on its grammatical category and associated linguistic features, the morphological engine applies morphophonological rules and affixation patterns to derive possible inflected forms. These forms are stored together with their grammatical annotations, creating a large morphological inventory linked directly to dictionary entries.

The verification module subsequently scans a dynamically expanding corpus to identify attested occurrences of generated forms. Attestation counts allow the system to distinguish between frequently occurring forms, rare but attested forms, and forms that are theoretically predicted but not observed.

When a user searches for a word form, the system traces it back to the base lemma, retrieves its grammatical analysis, and displays corresponding corpus examples. Dictionary entries, morphological modeling, and corpus data thus become mutually reinforcing components of a unified linguistic resource.

3.3 Annotation and Verification

Within the generate–verify framework, annotation and verification play complementary roles. Annotation ensures internal linguistic consistency by linking each generated form to its grammatical structure, while verification provides empirical grounding.

Corpus attestation thus serves as an indicator of reliability. Forms that occur frequently can be considered well-established elements of the language, whereas unattested forms may reflect either rare constructions, corpus gaps, or algorithmic overgeneration. Rather than discarding them, they are retained as candidates for further validation through future expanded corpus coverage or community feedback. To resolve the issue of non-standard orthography in digital texts, all corpus examples and generated word forms pass through the normalization pipeline (Section 4.1) prior to any attestation matching.

This dual mechanism allows for a balance of linguistic completeness with empirical validation. Morphological rules ensure that the full structural

potential of the language is represented, while corpus evidence and speaker contributions progressively refine this living resource.

By transforming descriptive grammatical knowledge into a data-generating computational process, the generate–verify framework helps overcome the circular dependency between linguistic resources and language technologies. Instead of waiting for large, annotated corpora before developing computational tools, the framework allows lexical resources, corpora, and linguistic models to grow side by side through iterative interaction.

4 Architecture and Morphological Generation

AvarLab is implemented as a relational system centered on the `Entry` model, which stores lemmas together with linguistic metadata such as orthography, transcription, part of speech, grammatical class, and glosses. Dependent models capture the inflectional structure of different parts of speech (e.g., NounCase, VerbForm, AdjectiveCase) and are linked to the base entry via foreign keys, enabling full paradigm generation while maintaining referential integrity.

The initial lexicon was seeded via OCR and manual digitization of a Russian-Avar dictionary (Alikhanov, 2003). The morphological rules driving generation are implemented as forward-only procedural python scripts manually engineered from descriptive grammars (Alekseev et al., 2012; Forker, 2017; Khangereev, 2011). Because this architecture is procedural rather than static, the generators are constantly updated. As new linguistic features are observed, or as integration with the corpus highlights specific structural constraints, the underlying python logic can be dynamically patched to refine the paradigms and resolve algorithmic overgeneration.

PostgreSQL indexing supports search across both Cyrillic and normalized forms, while flexible grammatical attributes such as tense, aspect, or polarity are stored in structured fields. A unified API layer, implemented with the Django REST Framework (Django REST Framework, n.d.), serves both the web platform and external interfaces.

4.1 Orthographic Normalization and Transcription

Digitizing Avar requires systematic orthographic normalization due to inconsistencies in representing the palochka letter (I) and digraphs (e.g., гъ, къ, кI, хъ, гI, кь). The normalization pipeline standardizes all character variants to Unicode forms and treats digraphs as atomic units during generation and search. Users often substitute the palochka (I) with visually similar characters such as “1”, “I”, or “l”. All variants are automatically converted to the canonical Unicode form U+04C0 for palochka.

A complementary rule-based transcription system converts normalized Cyrillic into IPA representations, derived from authoritative descriptions of Avar phonology (Alekseev et al., 2012; Forker, 2017), establishing a foundation for potential ASR and TTS applications. Additionally, a phonotactic validation module, derived from analysis of current 29,197 dictionary entries, catalogs 130 attested onset cluster types and 139 coda cluster types. A syllabification algorithm based on the Maximal Onset Principle segments any Avar word into syllables, feeding both the UI display and ML training data exports.

4.2 Automatic Annotation

An automatic annotation layer integrates linguistic heuristics with structural inference, enabling large-scale annotation without manually labeled corpora. Because most existing Avar resources provide Russian glosses, semantic information is inferred via cross-lingual alignment. Specifically, we use Russian morphological analysis (e.g., pymorphy3, a maintained fork of the analyzer described by (Korobov, 2015)) and fastText semantic embeddings (Grave et al., 2018) to map Russian lexical features, such as animacy or abstractness, to Avar grammatical classes and morphological constraints.

For verbs, argument structure and transitivity are inferred from the syntactic behavior of the Russian gloss and further refined through Avar class morphology. In addition, verbal argument structure and transitivity are automatically populated in the database by analyzing syntactic patterns across a pre-tagged corpus. This approach enables preliminary automatic inference of valency patterns from corpus evidence, reducing the amount of manual annotation required.

4.3 Morphological Generation

AvarLab’s morphological generator formalizes Avar inflectional morphology through a modular, rule-based system. To maintain structural consistency, the current generation engine strictly models the standard literary Avar dialect (based on the

Khunzakh variety), providing a stable baseline before accommodating the language's extensive regional variations. Each part of speech is handled by a dedicated generation module containing affixation rules, morphophonological transformations, and exception lists. The process consists of four steps:

1. Retrieve lemma and grammatical features. Paradigm selection is strictly deterministic: the assigned Part-of-Speech and metadata trigger the generation associated with that POS.

2. Apply affixation rules to produce inflected or derived forms.

3. Execute morphophonological adjustments (vowel deletion, consonant alternation, assimilation).

4. Store results with tags (e.g., case, number, tense).

All rules are linguistically interpretable, derived from descriptive grammars (Alekseev et al., 2012; Forker, 2017; Magomedkhanov et al., 2018), and validated through corpus evidence.

4.3.1 Nouns

Avar distinguishes four grammatical core cases: nominative/absolutive, ergative, genitive, dative/instrumental; and approximately twenty local cases organized into five positional series: on/over, near, inside/among, under/beneath, and inside a hollow object (Alekseev et al., 2012; Forker, 2017). Each positional series contains four directional subtypes: locative, allative, ablative, and perlative, yielding between 20 and 72 distinct case forms per noun (singular, possibly singular-alternative, and plural). Generating these paradigms requires resolving the two-stem principle: all indirect cases are built upon an oblique stem that frequently undergoes highly irregular morphophonological changes from the nominative root (vowel ablaut, syncope, or epenthesis). The engine algorithmically predicts these oblique stems across seven distinct structural types before affixing case endings.

The system handles Avar's contextual gender dualism at the database level. For human-referent nouns that can act as either male or female depending on context (e.g., устар "teacher"), the platform splits them into distinct Class I and Class II entries. This architectural decision enables the generator to assign the correct gender-specific ergative cases (устарас "teacher.CLI.ERG" vs. устаралъ "teacher.CLII.ERG").

Plural formation follows multiple morphophonological strategies conditioned by phonological shape, etymology, and semantic class. Irregular and suppletive nouns are handled through an exception table overriding rule-based output. Abstract nouns, typically resisting pluralization, are automatically detected via suffixes.

The engine also includes a dedicated Russian loanword declension module that correctly prevents native Avar phonological rules (such as vowel ablaut and high-vowel dissimilation) from applying to borrowings (e.g., correctly generating трактораль, tractor.ERG, instead of the incorrect native-rule form тракторуца).

4.3.2 Adjectives

Avar adjectives agree with head nouns in class and number, and inflect for case when used nominally (Alekseev et al., 2012; Forker, 2017). For standardization, adjective lemmas are normalized to the Class III form ending, serving as the default base, from which Class I, Class II, and plural are derived automatically.

Because Avar adjectives can function as nouns when substantivized, the engine generates full substantive declension paradigms as well. Furthermore, derived adjectives are linked to their base nouns or verbs (e.g., меседилаб "golden" → месед "gold"), tracking the derivational history of the word family.

4.3.3 Verbs

The implementation of verbal morphology in AvarLab addresses two interrelated challenges. First, the classification of verbs as class-based or non-class-based. Second, the generation of complete morphological paradigms for each verb. Given Avar's rich verb morphology, particularly the interplay between class agreement, tense-aspect forms, and participial constructions, an automated system was designed to balance accuracy with efficiency.

The generator also derives masdars automatically and links them bidirectionally to their source verbs, preserving their role in analytical tense formation.

Class-based verbs are identified using a combination of explicit listings and rule-based detection. A curated list, compiled from the literature (Alekseev et al., 2012; Forker, 2017; Magomedkhanov et al., 2018; Khangereev, 2011) and native speaker consultation, included verbs with embedded class markers in the root. These verbs are automatically tagged as class-based.

Beyond classification, a rule-based engine generates the full range of Avar synthetic verb forms. For each entry, it derives its masdar, simple past tense, simple future tense, constative tense, participles, and adverbial participles. All forms are generated in both affirmative and negative variants. Class-based verbs are inflected with the appropriate prefixes and suffixes according to their grammatical class, while irregular verbs such as букIине ("to be") and ине ("to go") are handled via explicit exceptions to ensure their unique forms are correctly represented.

Analytical verb constructions are generated dynamically. Storing every possible combination of main verbs and auxiliaries would unnecessarily bloat the database. Therefore, by modeling these multi-word constructions computationally on the fly, the system preserves strict database normalization and scaling efficiency while still allowing complex, multi-word queries over the corpus. This dynamic multi-word modeling establishes a critical technical foundation for future development of advanced POS tagging algorithms and UD treebanks.

4.3.4 Multi-Word Expressions and Other Parts of Speech

Notably, over 52% of the lexicon (14,429 entries) consists of multi-word expressions, reflecting Avar's rich phraseological structure. Because such a high density of multi-word expressions is typically a challenge for standard tokenizers, we automatically categorize them into specific subtypes, including light verb constructions (2,616), collocations (1,931), compound terms (927), and true idioms. Multi-word expressions are stored as unified lexical entries with explicit relational links to their base components, and matched in the corpus via a multi-word scanner with strict boundary detection.

The generator also covers pronouns, numerals, adverbs, postpositions, conjunctions, and interjections through smaller dedicated modules. These modules capture case, agreement, ambiguity, and structural governance where relevant, extending broad part-of-speech coverage beyond the noun, adjective, and verb systems described above.

5 The Data: Interlinking the Dictionary and Corpus

AvarLab represents a comprehensive multi-source collection. The platform integrates two complementary corpora in a hierarchical document-sentence architecture. The first is a monolingual Avar corpus derived from AvarCorpora (Volina, 2023), Telegram data (Telegram, n.d.), literature, Wikipedia (Wikipedia contributors, n.d.), educational materials, and others. The second is a trilingual Avar–Russian–English corpus built from dictionary imports, academic work, literature, folk texts, and user contributions.

The processing pipeline performs normalization, automated sentence segmentation, language detection filtering, and quality control measures to ensure data integrity. All corpora feed the same verification module and support attestation counting, lemma retrieval, and example extraction.

The current scale of AvarLab is summarized in Table 1, reflecting the integration of diverse literary, journalistic, and folkloric sources.

Category	Metric	Count
Lexicon	Total Lemmas	14,768
	Multi-Word Expressions	14,429
	Total Lexical Units	**29,197**
Morphology	Generated Inflected Forms	1,026,668
	Corpus Attested Forms	76,295
Corpus	Monolingual Sentences	296,228
	Trilingual Segments	18,680
	Total Source Documents	684

Table 1: Quantitative Overview of the AvarLab Ecosystem.

Future development plans include the integration of oral corpus data, historical texts, systematic inclusion of regional variants, and expansion into specialized domains to achieve broader coverage across time, space, and register.

5.1 Dictionary–Corpus Integration

The integration creates a feature for the end user: when a user searches for a highly inflected Avar word form, the Morphological Engine traces the morphological path back to the base lemma. The platform then simultaneously retrieves the dictionary definition and real, POS-tagged sentence examples from the corpus.

Searching supports orthographic normalization, fuzzy matching, lemma search, and keyword-in-context retrieval. Attested forms are linked to source sentences through an Example table, enabling direct access to real usage contexts.

5.2 Automatic POS Tagging and Annotation

Unlike conventional NLP pipelines that rely on manually annotated corpora to train POS taggers,

AvarLab adopts a dictionary-driven tagging approach. Instead of learning morphological patterns from annotated text, the system derives them directly from the morphological generator. The tagging workflow follows a reversed pipeline: Dictionary → Morphological Generator → WordForm Database → Corpus Tagging (see Fig. 1).

This "dictionary-driven" tagging allows for the rapid generation of large-scale silver-standard datasets, which can be exported for training neural POS taggers and language models. This workflow shifts the human role from manual labeling to verification, significantly reducing the time required to produce gold-standard corpora for Avar NLP.

To address the inherent morphological syncretism of Avar (e.g., distinguishing between visually identical case forms), the tagging pipeline incorporates a layer of contextual syntax rules. By applying pattern-matching heuristics to POS-tagged sentences, such as identifying strict Ergative-Nominative-Verb valency frames or adjacent Genitive-Noun pairs, the system actively disambiguates syntactic roles for high-frequency constructions. While these heuristics significantly reduce false positives during corpus attestation, completely resolving all structural ambiguity requires transitioning from morphological labeling to full dependency parsing. Consequently, this progressive formalization of Avar's structural syntax lays the concrete groundwork for automated parsing conforming to Universal Dependencies (UD).

5.3 Corpus-Based Dictionary Expansion

To support lexicon growth, AvarLab implements a dictionary coverage detection pipeline that identifies lexical items present in the corpus but absent from the dictionary. A baseline vocabulary set is constructed from all lemmas and generated inflected forms, and corpus tokens not present in this set are flagged as candidate lexical gaps. These candidates are ranked by frequency and then presented for human validation, enabling corpus-driven expansion of the dictionary.

5.4 System Evaluation and Results

The cumulative results of the AvarLab generate–verify framework are summarized in Table 1. By integrating rule-based morphological generation with corpus-driven verification, the system has achieved unprecedented scale for a Northeast Caucasian language, providing over 1 million inflected forms for 14,768 lexical units.

Although the overall verification rate for generated forms is 7.4%, this largely reflects Avar's high morphological density rather than a system weakness. The system achieves an average Part-of-Speech tagging coverage of 65.8% across the corpus, a significant baseline for a morphologically rich low-resource language. Nouns and adjectives dominate the lexicon but occur across a broad range of rare localized case forms, whereas verbs show higher verification rates due to their central syntactic role. Closed classes such as pronouns and adverbs exhibit the highest attestation levels. These results show that the generate–verify framework makes the missing-data problem explicit by generating valid paradigms beyond current corpus coverage.

6 Community Participation and Data Accessibility

At the time of writing, the community participation features are fully implemented, though the public release is forthcoming. The platform is designed to engage a diverse user base, including native speakers of various backgrounds, Avar language learners, educators, and linguists. While participation statistics are not yet available, establishing this moderation and contribution pipeline is a critical prerequisite for sustainable, community-driven resource expansion. Future work will evaluate user engagement and contribution patterns once the platform is deployed.

6.1 Community Contribution and Validation

Community participation is central to AvarLab's design. The platform provides multiple entry points for users to engage in collaborative validation and enrichment of the language resource. Users can submit new entries, suggest corrections via the web form or a bot, and upload pronunciation recordings. Forms that receive ≥ 10 positive votes are automatically marked as community verified, transitioning them from silver-standard generated data to gold-standard human-validated annotations.

Moderation occurs through an administrative dashboard where editors can review flagged items, merge duplicates, or approve community-submitted entries. This workflow balances collaboration with curated linguistic oversight, keeping the dictionary inclusive yet academically reliable.

6.2 User Interaction and Interface Design

The interface is designed for both specialists and non-specialists. Search supports orthographic normalization, fuzzy matching, and retrieval of inflected forms, allowing users to move from surface forms to lemmas and corpus examples. Access is also provided through a bot interface linked to the same API, enabling lexical lookup, example retrieval, and error reporting outside the web platform.

6.3 Training Data Export for NLP

AvarLab supports export of annotated data in formats such as JSON, CoNLL-U, and spaCy-compatible datasets, with optional train/validation/test splitting. This makes the platform not only a reference resource but also a source of structured data for downstream NLP development.

7 Discussion

AvarLab demonstrates that sustainable digital infrastructure for morphologically complex, low-resource languages can be developed even under severe data scarcity. By reversing the conventional corpus-first paradigm, the generate–verify framework allows linguistic modeling, corpus expansion, and community participation to develop in parallel. Instead of requiring large annotated corpora as a prerequisite, the system transforms descriptive grammatical knowledge into a data-generating computational process.

By encoding Avar morphology in a machine-readable form, AvarLab turns descriptive linguistic rules into active computational resources. Generated paradigms provide large-scale morphological coverage, while corpus-based verification ensures empirical grounding. This verification loop functions as a dynamic quality-control mechanism: attested forms strengthen the reliability of the model, while unattested or ambiguous forms are returned for further validation through corpus expansion and community input.

Beyond its technical contribution, AvarLab establishes an architectural blueprint for how computational infrastructure can support participatory language documentation. By allowing speakers to contribute lexical entries, usage examples, and pronunciation data, the platform decentralizes lexicographic practice and aligns digital resource development with principles of community-driven language preservation.

More broadly, the AvarLab architecture illustrates a scalable strategy for developing computational resources for morphologically rich, low-resource languages. Integrating morphological generation, corpus verification, and participatory validation provides a pathway toward building NLP-ready datasets and language technologies in contexts where traditional data-driven approaches remain infeasible. Finally, the scale of this generate-verify loop provides a feedback mechanism for morphological theory itself. Initial corpus verification has begun to highlight specific areas where traditional descriptive grammars over-generate, such as deeply nested spatial cases that are theoretically permissible but empirically absent from the 300,000-sentence corpus. Quantifying these empirical gaps to refine formal constraints on Avar productivity represents a promising avenue for future linguistic research.

8 Conclusion and Future Work

We presented the generate–verify framework, a methodology that integrates computational morphology, corpus linguistics, and community collaboration to build sustainable infrastructures for low-resource languages. Using Avar as a case study, AvarLab shows how rule-based generation, corpus verification, and community validation can form a scalable system that evolves with data and participation.

AvarLab currently provides comprehensive morphological coverage across all parts of speech, generating over 1 million inflected forms for 14,768 lexical entries withina relational database architecture. Beyond static documentation, it functions as a living linguistic resource that links lexical entries, corpus evidence, and speaker contributions.

Future work will focus on expanding the corpus with spoken and dialectal data, training downstream NLP models on AvarLab-generated datasets, and adapting the framework to other morphologically rich Northeast Caucasian languages. The public release of AvarLab will also enable empirical analysis of community participation and collaborative lexicon growth.

Ultimately, this project demonstrates that preserving linguistic diversity in the digital era is not only feasible but sustainable when technology and community act together.

Limitations

AvarLab currently relies primarily on rule-based modeling and therefore inherits the limitations of the descriptive resources on which these rules are based. Although corpus verification helps ground generated forms empirically, corpus coverage remains uneven across genres and registers, and many valid but rare forms remain unattested. Some annotation procedures, including class inference and valency detection, are heuristic and have not yet been evaluated against a gold-standard benchmark, as no such comprehensive dataset currently exists for Avar. Furthermore, while the system employs contextual syntax rules to mitigate morphological syncretism for high-frequency patterns (Section 5.2), resolving all structural ambiguity to eliminate false positives in corpus attestation requires the completion of a full dependency parser, which remains an area of active development. In addition, while community participation features are implemented, they have not yet been evaluated under public deployment. Future work will focus on broader corpus diversification, intrinsic evaluation of annotation accuracy, and user-based validation studies.

Ethical Considerations

This work is motivated by the need to support the digital representation of an under-resourced language and to develop computational tools that are useful both for research and for the speaker community. The platform is designed to support community contribution while preserving editorial oversight for quality control. User-contributed data, including lexical suggestions and audio recordings, will require clear consent and moderation policies upon public release. We also note that automated generation and annotation may introduce errors; therefore, outputs should be treated as computational analyses subject to revision rather than as authoritative linguistic judgments.

Acknowledgments

Kebed Zagidov and Thomas Brochhagen are funded by grant EVOSIG PID2024-162668NA-I00, funded by MICIU/AEI/10.13039/501100011033 and by "ERDF A way of making Europe". TB is also funded by the Ministerio de Ciencia, Innovacion, y Universidades, the Agencia Estatal de Investigacion, and the European Social Fund Plus (ref. RYC2023-045215-I MCIU/AEI/10.13039/501100011033).

References

M. E. Alekseev, B. M. Ataev, M. A. Magomedov, M. I. Magomedov, G. I. Madieva, P. A. Saidova, and D. S. Samedov. 2012. . Aleph, Makhachkala. [The contemporary Avar language].

S. Z. Alikhanov, editor. 2003. *Russian–Avar Dictionary: Over 40,000 Words*. Dagestan Scientific Center, Russian Academy of Sciences, Makhachkala.

Gregory D. S. Anderson. 2011. Language hotspots: What (applied) linguistics and education should do about language endangerment in the twenty-first century. *Language and Education*, 25(4):273–289.

B. T. S. Atkins and Michael Rundell. 2008. *The Oxford Guide to Practical Lexicography*. Oxford University Press, Oxford.

Mark Davies. 2008. The corpus of contemporary american english (COCA). https://www.english-corpora.org/coca/. Corpus.

Django REST Framework. n.d. Django rest framework. https://www.django-rest-framework.org/. Accessed: 2026-03-26.

First Peoples' Cultural Council. 2018. Firstvoices: Indigenous language archiving and teaching platform. https://www.firstvoices.com/. Platform documentation.

Diana Forker. 2017. Avar: Grammar sketch. In Michael Daniel, Timur Maisak, and Ekaterina M. Vinogradova, editors, *The Oxford Handbook of the Languages of the Caucasus*. Oxford University Press, Oxford.

Jim Giles. 2005. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293. Association for Computational Linguistics.

M. D. Khangereev. 2011. *Paradigmaticheskaya sistema glagola v avarskom yazyke*. Dagestan State University, Makhachkala. [ ] [The paradigmatic system of the verb in the Avar language].

András Kornai. 2013. Digital language death. *PLoS ONE*, 8(10):e77056.

Mikhail Korobov. 2015. Morphological analyzer and generator for Russian and Ukrainian languages. In *Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue"*, volume 14(21), pages 320–332.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Tommi A. Pirinen, and Miikka Silfverberg. 2013. HFST—a system for creating NLP tools. In Alexander Gelbukh, editor, *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*, pages 53–71. Springer.

Living Tongues Institute for Endangered Languages. n.d. Language hotspots. https://livingtongues.org/language-hotspots/. Accessed: 2026-03-27.

M. M. Magomedkhanov, Kh. M. Bechedova, and R. M. Yusupova. 2018. *Samouchitel' avarskogo yazyka*. Epokha Publishing House, Makhachkala. [ ] [Self-study guide of the Avar language].

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2010.12316*.

Christopher Moseley, editor. 2010. *Atlas of the World's Languages in Danger*, 3rd edition. UNESCO Publishing, Paris.

Sjur N. Moshagen, Tommi A. Pirinen, and Trond Trosterud. 2014. Building an open-source development infrastructure for language technology projects. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2433–2440. European Language Resources Association (ELRA).

Wilhelmina Nekoto and 1 others. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160. Association for Computational Linguistics.

John Sinclair. 1991. *Corpus, Concordance, Collocation*. Oxford University Press, Oxford.

Telegram. n.d. [hakikat]. https://t.me/hakikat_gazeta. Telegram channel.

Volina. 2023. Avarcorpora. https://huggingface.co/datasets/volina092/avarCorpora. Dataset.

Wikipedia contributors. n.d. Avar Wikipedia API. https://av.wikipedia.org/w/api.php. Dataset.