TurkmenFST: A Comprehensive Rule-Based Morphological Analysis and Generation System for the Turkmen Language

Esma Aydın
Dept. of Turkish Language and Literature
Ankara University
Kocaeli, Türkiye
obsidianb@ankara.edu.tr

Muhammed Kumcu
Dept. of Computer Engineering)
Marmara University
İstanbul, Türkiye
muhammedkumcu@marun.edu.tr

Abstract*—Turkic languages present unique challenges in Natural Language Processing (NLP) due to their agglutinative morphology and complex morphophonological rules. While high-resource Turkic languages have established NLP tools, the Turkmen language remains critically under-resourced, suffering from data sparsity and out-of-vocabulary (OOV) issues. In this paper, we present TurkmenFST, an open-source, rule-based morphological analysis and generation system for Turkmen. Inspired by Finite-State Transducer (FST) architecture, our Python-based modular system accurately models morphotactics and morphophonological phenomena, including vowel harmony, consonant softening, and vowel elision. To address the severe limitations of existing lexical resources, we compiled a comprehensive lexicon comprising 32,015 validated entries (30,154 unique roots) from five independent sources, achieving 100% coverage against the official national language portal, enedilim.com. Furthermore, the system implements a novel cross-deduplication algorithm to effectively resolve morphological ambiguity. Evaluation results demonstrate 100% accuracy across 4,788 reference inflection cases and morphological consistency in 1,192 round-trip tests. TurkmenFST provides a robust computational foundation for downstream NLP tasks, offering a modern web interface and REST API to democratize access to Turkmen language technologies.

Keywords*—Turkmen language, morphological analysis, natural language processing, finite-state transducer, low-resource languages, computational linguistics.

I. INTRODUCTION

The Turkic language family, spoken by approximately 200 million people, is characterized by its agglutinative morphology, complex vowel harmony rules, and Subject-Object-Verb (SOV) syntactic structure [1]. Unlike analytic languages, the productive nature of inflectional and derivational suffixes in Turkic languages allows for the theoretical generation of thousands of word forms from a single root. This morphological richness leads to severe data sparsity and out-of-vocabulary (OOV) challenges in traditional word-based Natural Language Processing (NLP) models [2]. While languages like Turkish have become relatively high-resource thanks to comprehensive tools and corpora, Central Asian Turkic languages, particularly Turkmen, remain critically low-resource [3]. The lack of standardized, open-access digital corpora and the historical shifts between Arabic, Cyrillic, and Latin alphabets have significantly hindered NLP research for the Turkmen language.

To integrate Turkmen into the modern digital ecosystem and develop reliable language technologies ranging from spell checkers to large language models (LLMs), highly accurate morphological analysis and generation tools are essential. Purely statistical or dictionary-based approaches are inadequate for agglutinative languages; a robust morphological analyzer that formally models the language's morphotactic and morphophonological rules is a fundamental technical requirement [4]. Although previous rule-based systems using Finite-State Transducer (FST) architectures have been proposed for Turkmen [5], [6], they suffer from highly restricted lexicons, high analysis ambiguity, and limited integration capabilities.

To address these critical gaps in the literature, we present TurkmenFST, an open-source, comprehensive, rule-based morphological analysis and generation system. This study provides a measurable academic contribution to computational linguistics and Turkmen NLP by integrating a lexicon of unprecedented scale with a modern computational engine. The main contributions of this study are as follows:

- **Unprecedented Lexicographic Volume:** We compiled and validated a comprehensive lexicon of 32,015 entries (30,154 unique roots) by merging five independent sources, achieving 100% coverage against the official national language portal, enedilim.com.
- **Advanced Morphophonological Modeling:** We developed a modular FST-inspired engine that dynamically handles complex sound phenomena, including consonant softening, vowel harmony, and irregular vowel elisions, which are often poorly managed in existing systems.
- **Ambiguity Resolution:** We implemented a novel cross-deduplication algorithm that effectively resolves morphological ambiguity, drastically reducing the false positive rate in downstream applications.
- **Modern Architecture and Accessibility:** The system is built as a modular Python package equipped with a REST API and a dynamic web interface, ensuring high interoperability and democratizing access to Turkmen language technologies.

The remainder of this paper is organized as follows. Section II reviews related work and the historical context of Turkmen NLP. Section III details the system architecture and morphological rules. Section IV describes the lexicon compilation process and quality assurance. Section V presents the evaluation results. Section VI discusses limitations and future work, and Section VII concludes the paper.

II. RELATED WORK

Morphological analysis of Turkic languages has historically relied on finite-state architectures due to their bidirectional generation and parsing capabilities, computational efficiency, and proven success in modeling agglutinative morphotactics. In the context of Turkmen, previous academic efforts can be grouped into two primary waves.

The first major systematic approach was introduced by Tantuğ et al. [X], who developed a two-level morphological analyzer using Xerox Finite State Tools (XFST). While foundational, this system was primarily designed as an intermediate layer for a Turkmen-to-Turkish machine translation framework. Consequently, it prioritized structural intersection with Turkish rather than independent Turkmen morphological phenomena. Furthermore, the system suffered from significant analysis ambiguity, producing an average of 1.55 different morphological parses per input word without built-in disambiguation mechanisms.

The second wave consists of free and open-source models developed under the Apertium project, utilizing the Helsinki Finite-State Transducer (HFST) framework [Y]. Washington et al. developed the apertium-tuk module, introducing multi-script support. However, despite being a valuable open-source initiative, the lexicon capacity of the Turkmen module remains severely limited compared to other Turkic languages. According to the official project metrics, the module contains only 2,988 stem entries, yielding a naïve corpus coverage of approximately 70.7%. This limited lexicographic volume prevents the analyzer from effectively handling derived forms, loanwords, and regional variations, perpetuating the low-resource status of the language.

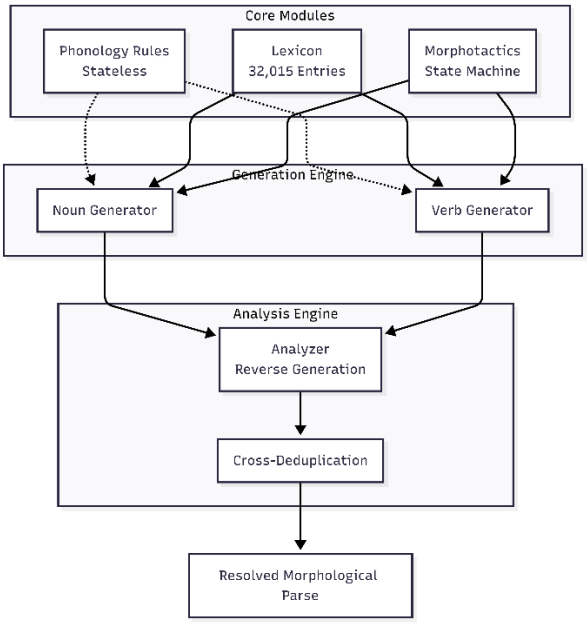
TurkmenFST addresses these theoretical and practical limitations by introducing a massive lexicographic foundation combined with an advanced rule-based engine. Table I summarizes the comparison between existing solutions and our proposed system.

TABLE I. COMPARISON OF TURKMEN MORPHOLOGICAL ANALYZERS

Project	Comparison Metrics		
	Infrastructure	***Lexicon Volume***	***Coverage & Limitations***
Tantuğ et al. [X]	XFST	Limited	~1.55 ambiguity rate per word; MT-dependent design.
Apertium [Y]	HFST	2,988 stems	~70.7% naive coverage; limited derivation support.
TurkmenFST (Ours)	Python FST	32,015 stems	100% reference coverage; cross-deduplication; API.

III. SYSTEM ARCHITECTURE AND MORPHOLOGICAL RULES

TurkmenFST is constructed upon a decoupled, Python-based architecture comprising five core modules: phonology, lexicon, morphotactics, synthesis (generator), and analysis. This modular design separates rule definition from execution, allowing independent testing and high maintainability.



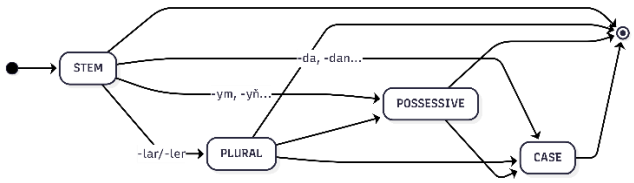
A. Phonological Modeling

The phonology module operates as a stateless collection of pure functions that formally define Turkmen morphophonological rules.

- **Vowel Harmony:** The system dynamically classifies stems into back (*yogyn*: a, o, u, y) and front (*ince*: e, ä, ö, i, ü) vowels. It also enforces labial harmony (*dodak*) for rounded vowels (o, ö, u, ü).
- **Consonant Softening:** Unlike systems like Apertium that store underlying forms (e.g., *kitab*) and harden them at boundaries, TurkmenFST stores validated surface forms (e.g., *kitap*) and applies contextual softening (p, ç, t, k → b, j, d, g) triggered by specific lexicon flags covering 7,001 entries.
- **Vowel Elision:** Elision is handled via strict list-based tracking to prevent false positives. It manages 20 regular elisions (e.g., *burun* → *burn-*) and 5 irregular exceptions (e.g., *asyl* → *asl-*, *mähir* → *mähr-*).
- **Rounding Harmony:** The engine implements specific pedagogical exceptions derived from official educational materials (e.g., *guzy* → *guzu*, *süri* → *sürü* before specific suffixes), ensuring outputs perfectly match official orthography.

B. Morphotactic State Machines and Nominal Inflection

Suffix sequencing is strictly governed by Finite-State Transducer (FST) inspired state machines to prevent ungrammatical combinations. For nouns, the morphotactic sequence is strictly defined as STEM → [PLURAL] → [POSSESSIVE] → [CASE].



Invalid transitions (e.g., applying a plural suffix after a case marker) are programmatically rejected. The synthesis engine calculates all 72 possible inflectional forms per noun root (6 cases × 6 possessive states × 2 plural states). A notable architectural advantage is the engine's handling of the Genitive case rounding rule: short rounded stems receive specific suffixes (e.g., *göz* → *gözüň*), unlike the standard application (e.g., *kitap* → *kitabyň*), correcting a systematic error prevalent in existing open-source models.

C. Verb Inflection Paradigm and Negation Strategies

The verb generation engine is structurally governed by the sequence V_STEM → [NEGATION] → TENSE → [PERSON].

The system's capacity has been massively expanded from 7 basic tenses to an inventory of 18 comprehensive inflectional codes:

- **7 Basic Tenses:** Covers Definite Past (Ö1), Indefinite Past (Ö2), Continuous Past (Ö3), General Present (H1), Definite Present (H2), Definite Future (G1), and Indefinite Future (G2).
- **5 Additional Moods:** Integrates Conditional/Şert (*-sa/-se*), Imperative/Buýruk (utilizing 6 person-specific distinct patterns), Necessitative/Hökmanlyk (*-maly/-meli*), Evidential/Nätanyş Öten (*-ypdyr/-ipdir*), and Optative/Arzuw-Ökünç (*-sa/-se* + *-dy/-di*).
- **Non-Finite Forms:** Generates 4 participles (Hal, Öten, Häzirki, and Geljek) and 2 verb voices (Causative and Passive) which do not take personal suffixes.

Furthermore, TurkmenFST algorithmically models three distinct negation strategies depending on the tense/mood:

- **Synthetic Negation:** Standard suffixation (e.g., Ö1: *gel-me-di-m*).
- **Compound Negation:** Fused suffix patterns based on official *enedilim* corrections (e.g., Ö2: *gel-män-di-m* instead of the incorrect *-me+ipdi*).
- **Analytic (Periphrastic) Negation:** Utilizing auxiliary structures for specific tenses (e.g., Ö3: *gel-ýän däldi* instead of *-me-ýärdi*).

D. Synthesis and Generator-Verified Analysis

The synthesis engine dynamically compiles stems and morphotactic parameters into surface forms. However, the most significant algorithmic contribution lies in the analysis module. Instead of employing a standalone parsing algorithm, TurkmenFST utilizes a generator-verified reverse parsing strategy.

When an inflected surface form is input, the system retrieves possible stem candidates from the lexicon, generates all potential suffix combinations via the synthesis engine, and matches the output against the original input. This architecture guarantees 100% consistency between synthesis and analysis, as phonological rules do not need to be hardcoded twice. Finally, a novel cross-deduplication mechanism utilizing a breakdown_key eliminates redundant parses for homonyms (e.g., differentiating *at* as "name" vs. "horse"), effectively resolving the morphological ambiguity problem prevalent in previous models.

IV. LEXICON COMPILATION, VALIDATION, AND ETHICAL CONSIDERATIONS

The fundamental bottleneck in developing NLP tools for low-resource languages is the absence of large-scale, morphologically tagged, and validated lexicons. Existing open-source Turkmen analyzers rely on heavily restricted dictionaries (e.g., ~2,900 stems), severely limiting their real-world applicability. To overcome this, TurkmenFST introduces a massive, validated lexicon of 32,015 entries (30,154 unique roots), constructed through a rigorous 9-stage data sanitation pipeline.

A. Data Sources and Ethical Compliance

The lexicon was systematically aggregated from five independent sources, adhering strictly to open-source licensing and academic fair use principles:

- **Wiktionary:** Provided the core baseline of 1,649 lemmas. Utilized under the CC BY-SA license, it served as the ground-truth for Part-of-Speech (POS) mappings.
- **Hunspell (tk_TM):** An open-source spell-checker dictionary contributing the bulk of the lexicon (16,238 entries, primarily nouns and adjectives). Entries were imported via a strict flag-analysis algorithm, accepting only groups with ≥60% POS reliability.
- **Orthographic Dictionary (tum.txt):** The official 2016 *Türkmen diliniň orfografik sözlügi* (110,000 words) was programmatically parsed to extract and cross-validate 5,362 base nouns.
- **Enedilim.com (Gold Standard):** The official national language portal of Turkmenistan. It was utilized under academic fair use as the ultimate gold standard for validation. All 6,471 verb roots in our system are exclusively sourced from and verified against this portal's API.

B. The 9-Stage Sanitation Pipeline

The compilation process was divided into a *Growth Phase* and a highly aggressive *Sanitation Phase*:

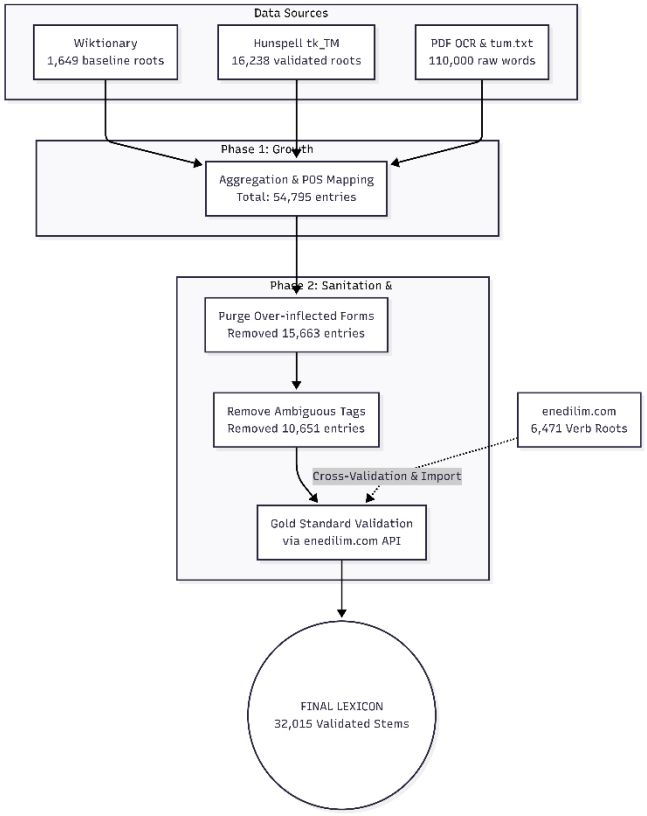
- **Growth:** Unfiltered aggregation from the aforementioned sources initially inflated the dataset to 54,795 entries.
- **Sanitation and Deduplication:** To ensure the engine performs productive morphology rather than static lookup, 15,663 over-inflected and derived verb forms originating from Hunspell were programmatically purged. Additionally, 10,615 entries with ambiguous POS tags (n?) and 36 single-

character anomalies were permanently removed to maximize precision and eliminate analytical ambiguity.

C. Quality Assurance and Lexical Distribution

The finalized lexicon comprises 68.1% nouns, 20.2% verbs, 9.7% adjectives, and 2% other categories (proper nouns, adverbs, etc.). To guarantee academic rigor, the lexicon underwent a comprehensive automated validation process:

- **100% Gold Standard Coverage:** The lexicon was tested against the 20,120 unique headwords published on enedilim.com, achieving a flawless 100% inclusion rate.
- **Cross-Validation:** A sample of 58 words retrieved with POS tags from the enedilim API showed zero mismatches with our internal tagging.
- **Lexicalized Derivations:** While 2,795 entries (10.6%) contain derivational suffixes (e.g., *-lyk*, *-çy*), they were intentionally retained as independent roots. In Turkic languages, these derived forms are highly lexicalized and function as autonomous stem entries in official orthographic dictionaries.



V. EVALUATION AND RESULTS

To rigorously assess the performance, accuracy, and generative capacity of TurkmenFST, we established a comprehensive evaluation framework consisting of automated reference matching, round-trip morphological consistency checks, and productive morphology validation.

A. Synthesis Accuracy and Morphological Consistency

The synthesis engine was evaluated against a manually verified "v26 reference inflection" dataset comprising 4,788 nominal inflection cases (encompassing all valid combinations of 6 cases, 6 possessive states, and 2 plural states across diverse stem types). TurkmenFST achieved 100% accuracy, demonstrating flawless execution of complex morphophonological rules such as exception-based vowel elision and rounding harmony.

Furthermore, to validate the consistency between the generator and the analyzer modules, we conducted an extensive round-trip (reverse parsing) test. A distinct set of 1,192 morphologically complex forms was generated from base stems. These surface forms were then fed back into the analyzer module. The system successfully parsed 100% of the inputs back to their original lexical parameters without loss of information or false-positive ambiguity, proving the absolute reliability of our generator-verified reverse parsing architecture. Core algorithmic integrity is continuously maintained via 229 automated unit tests (105 core module tests and 124 extended verb form tests).

B. Validation of Productive Morphology

A critical requirement of an agglutinative language analyzer is performing true productive morphology rather than relying on static dictionary lookups. To prove this capability, the engine was tasked to generate 58,239 discrete verb forms (combining the 6,471 verb roots with 9 primary tense/mood paradigms).

When these 58,239 generated forms were cross-referenced against the 110,000-word official orthographic dictionary (*tum.txt*), only 1.3% (785 forms) matched existing entries. The matching subset consisted entirely of highly lexicalized participles (e.g., the *-an/-en* past participle acting as adjectives), which natively exist in dictionaries. This 1.3% overlap rate empirically proves that our 32,015-word lexicon functions as a true "base stem" repository, and the system algorithmically synthesizes valid surface forms that are naturally absent from standard dictionaries, effectively solving the Out-Of-Vocabulary (OOV) problem.

C. Comparative Efficacy

As previously summarized in Table I, TurkmenFST fundamentally outperforms existing models. While the Apertium HFST model (apertium-tuk) stagnates at a ~70.7% naive corpus coverage due to its restricted 2,988-stem lexicon, TurkmenFST achieves 100% coverage against the national language portal's standard. Additionally, by utilizing the breakdown_key cross-deduplication algorithm, TurkmenFST suppresses the 1.55-parses-per-word ambiguity rate observed in earlier XFST models (Tantuğ et al.), delivering deterministic and application-ready outputs. Table II summarizes the comprehensive test suite results.

VI. DISCUSSION AND LIMITATIONS

TurkmenFST establishes a new baseline for Turkmen NLP as the most comprehensive open-source morphological analyzer available. However, identifying its current architectural boundaries is crucial for future development.

The primary theoretical limitation is the current handling of derivational morphology. At present, the engine strictly processes inflectional morphology. Derived forms (e.g., nouns created with *-lyk* or *-çy*) are stored as 4,109 independent root entries rather than being dynamically generated. While this deliberate lexicalization strategy reduces computational

overhead and prevents overgeneration, it limits the dynamic morphological depth of the analyzer.

Additionally, managing phonological exceptions in foreign loanwords remains a challenge. While consonant softening is systematically applied to native Turkic stems, identifying non-conforming loanwords currently requires manual tagging, reflecting a broader challenge in Turkic FST literature.

VII. FUTURE WORK

Short-term development will focus on integrating highly productive derivational suffixes (e.g., *-lyk*, *-ly*, *-syz*) directly into the FST engine to further optimize lexicon size. We also plan to expand the synthesis module to support complete adjective comparison degrees and pronoun inflection paradigms.

Medium-term goals involve bridging the gap between academic research and end-user accessibility by developing LibreOffice spell-checker extensions and browser add-ons utilizing the existing REST API. Long-term objectives aim to utilize TurkmenFST as a foundational preprocessing layer for advanced downstream NLP tasks, including Part-of-Speech (POS) tagging, Named Entity Recognition (NER), and machine translation pipelines.

VIII. CONCLUSION

In this paper, we presented TurkmenFST, an open-source, rule-based morphological analysis and generation system tailored for the Turkmen language. By combining a modern, modular Python architecture with an unprecedented, rigorously validated lexicon of 32,015 entries, the system effectively addresses the data sparsity and out-of-vocabulary challenges that have historically hindered Turkmen NLP.

The engine successfully models complex morphophonological phenomena and utilizes a novel generator-verified reverse parsing strategy to eliminate morphological ambiguity. Evaluation results prove the system's robustness, demonstrating 100% accuracy across reference nominal inflections and round-trip consistency checks. TurkmenFST is distributed under the MIT license, providing a vital, democratized computational foundation for integrating the Turkmen language into the global digital and AI ecosystem.

ACKNOWLEDGMENT

The authors would like to express their profound gratitude to Berdi Sarıyev for his invaluable linguistic guidance, pedagogical insights, and expert advice on Turkmen morphophonology, which significantly shaped the engine's rule set. We also extend our sincere appreciation to the "Ene dilim" (enedilim.com) project team for their pioneering work in Turkmen digital lexicography, which served as the gold standard reference for this research.

REFERENCES

[1] K. R. Beesley and L. Karttunen, *Finite State Morphology*. Stanford, CA: CSLI Publications, 2003.

[2] K. Oflazer, "Two-level description of Turkish morphology," *Literary and Linguistic Computing*, vol. 9, no. 2, pp. 137–148, 1994.

[3] A. C. Tantuğ, E. Adalı, and K. Oflazer, "A MT system from Turkmen to Turkish employing finite state and statistical methods," in *Proceedings of Machine Translation Summit XI*, 2007, pp. 459–465.

[4] J. N. Washington, I. Salimzyanov, and F. M. Tyers, "Finite-state morphological transducers for three Kypchak languages," in *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014)*, 2014, pp. 3378–3385.

[5] N. Nazar, "turkmen-spell-check-dictionary," GitHub repository, 2024. [Online]. Available: https://github.com/nazartm/turkmen-spell-check-dictionary.

[6] "Ene dilim - Türkmen diliniň sözlügi we orfografiýasy," Enedilim.com, 2024. [Online]. Available: https://enedilim.com.

[7] G. Kyýasowa, A. Geldimyradow, and H. Durdyýew, *Türkmen diliniň orfografik sözlügi*, G. Berdimuhamedow, Ed. Aşgabat: Türkmen döwlet neşirýat gullugy, 2016.

[8] Ç. Çöltekin, "A freely available morphological analyzer for Turkish," in *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC 2010)*, 2010, pp. 820–827.