SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 1 | January 2026 - Peer Reviewed Journal

A HYBRID FINITE-STATE AND RULE-BASED ARCHITECTURE FOR THE ANGIKA MORPHOLOGICAL ANALYZER GENERATOR

Alok Kumar[1*], **Manisha Kumari Deep**[2]
[1]*Department of Computer Science Engineering, YBN University Ranchi*
[2]*School of Computer Science and IT, YBN University Ranchi*

Article DOI: https://doi.org/10.36713/epra25947
DOI No: 10.36713/epra25947

ABSTRACT

Morphological analysis is a core component of natural language processing, particularly for mor- phologically rich and low-resource languages where data-driven approaches are often impractical. This paper presents a hybrid finite-state and rule-based morphological analyzer generator for Angika, an under-resourced Indo-Aryan language. The proposed system adopts a generator-oriented architecture that separates linguistic knowledge from core processing mechanisms, enabling modularity, extensibil- ity, and reuse across related languages. Finite-state transducers are employed to model regular inflec- tional morphology, while declarative linguistic rules handle irregular and constraint-sensitive phenomena that cannot be reliably captured through finite-state representations alone. The interaction between the two components is governed by a deterministic integration and conflict resolution strategy, ensuring predictable and interpretable analysis. Experimental evaluation demonstrates that the hybrid approach achieves higher coverage and accuracy than finite-state-only and rule-based-only baselines. The results confirm that a generator-based hybrid architecture provides an effective and scalable solution for mor- phological analysis in low-resource linguistic settings.

KEYWORDS: *Morphological Analyzer Generator, Finite-State Morphology, Rule-Based Morphology, Low-Resource Languages*

1 INTRODUCTION AND BACKGROUND

Morphological analysis constitutes a foundational component of natural language processing by enabling the systematic decomposition of surface word forms into lexical roots, affixes, and morphosyntactic features. For languages with rich inflectional morphology, accurate word-level analysis is indispensable for downstream tasks such as syntactic parsing, machine translation, and information retrieval, as it directly reduces lexical sparsity and improves structural generalization [9, 2]. Consequently, computational morphology continues to play a central role in NLP architectures, particularly in settings where higher-level models depend on linguistically informed representations.

Despite significant advances in data-driven and neural approaches, morphological processing in low- resource languages remains a persistent challenge. Statistical and neural models typically require large volumes of annotated data to achieve robust generalization, a requirement that is rarely satisfied for under- resourced languages [4, 3]. In such contexts, the absence of extensive corpora, incomplete lexical coverage, and high morphological productivity severely limit the applicability of purely data-driven techniques. As a result, linguistically motivated approaches that encode explicit morphological knowledge remain essential for reliable analysis in low-resource environments.

Finite-state morphology has emerged as one of the most influential formalisms for computational mor- phological analysis due to its formal rigor, efficiency, and bidirectional processing capabilities [2, 5]. Finite- State Transducers (FSTs) are particularly effective in modeling regular, concatenative morphological patterns and have been successfully applied across diverse languages. However, finite-state-only systems exhibit in- herent limitations when confronted with irregular inflection, morphophonemic alternations, and constraint-sensitive agreement phenomena that violate regularity assumptions [8]. Encoding such phenomena exclu- sively within finite-state models often leads to state explosion and reduced maintainability.

Rule-based morphological systems address these limitations by explicitly encoding linguistic constraints and exception patterns, thereby offering greater descriptive precision and transparency [1]. Nevertheless, when deployed in isolation, rule-based approaches suffer from scalability and efficiency issues, especially as the number of rules increases with morphological complexity. These complementary strengths and weak- nesses motivate the adoption of hybrid architectures that integrate finite-state mechanisms with rule-based components, enabling efficient handling of regular morphology while preserving the flexibility required to model irregular and language-specific patterns [6, 8].

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 1 | January 2026 - Peer Reviewed Journal

Angika, an Indo-Aryan language with productive inflectional morphology and limited computational resources, exemplifies a class of languages for which hybrid morphological processing is particularly well suited. The language exhibits largely concatenative nominal and verbal morphology alongside irregular forms and agreement-sensitive constructions that cannot be uniformly captured by finite-state transitions alone. From a computational perspective, Angika represents a low-resource scenario in which linguistically informed system design is essential due to the impracticality of corpus-dependent approaches. The present study restricts its scope exclusively to system-relevant morphological phenomena that directly inform ana- lyzer design and implementation.

The objective of this work is to develop a hybrid finite-state and rule-based morphological analyzer gen- erator that abstracts over language-specific details while maintaining linguistic adequacy and computational efficiency. Unlike conventional static analyzers, the proposed system is conceived as a generator framework, emphasizing modularity, extensibility, and reusability across related languages. By integrating finite-state processing for regular morphology with rule-based refinement for irregular and constraint-sensitive forms, the proposed architecture aims to provide a robust and scalable solution for morphological analysis in low- resource linguistic settings. The remainder of this paper presents the theoretical motivation, system design, implementation details, and empirical evaluation of the proposed approach.

2 RELATED WORK AND RESEARCH GAP

Computational morphology has been studied extensively through a variety of formalisms, with finite-state, rule-based, and hybrid approaches forming the core methodological paradigms. Finite-state morphology, particularly through the use of Finite-State Transducers (FSTs), has been widely adopted due to its formal elegance, computational efficiency, and bidirectional mapping between surface forms and lexical represen- tations [2, 5]. Such systems have demonstrated strong performance in modeling regular, concatenative mor- phological processes across a wide range of languages. However, finite-state-only approaches are inherently limited in their ability to capture irregular inflection, morphophonemic alternations, and constraint-sensitive agreement phenomena without introducing excessive state complexity or ad hoc extensions [8].

Rule-based morphological systems address these limitations by explicitly encoding linguistic knowledge in the form of handcrafted rules that model exceptions, contextual constraints, and language-specific patterns [1]. These systems offer high transparency and descriptive precision, making them particularly attractive for linguistically complex languages. Nevertheless, rule-based approaches scale poorly when applied in isola- tion, as increasing rule inventories lead to maintenance difficulties, reduced computational efficiency, and unpredictable interactions between rules [7]. Consequently, purely rule-driven systems are rarely sufficient for large-scale or extensible morphological processing.

Hybrid morphological architectures have been proposed to reconcile the complementary strengths of finite-state and rule-based approaches. In such systems, finite-state mechanisms typically handle regular morphological patterns, while rule-based components are employed to resolve irregular forms and constraint- driven phenomena [6, 8]. Empirical studies have shown that hybrid systems achieve improved coverage and robustness compared to single-paradigm models. However, most existing hybrid implementations are designed as language-specific analyzers, tightly coupled to a fixed set of linguistic descriptions, which limits their extensibility and reuse across related languages.

Morphological analysis in low-resource languages introduces additional challenges that further expose the limitations of existing approaches. Data-driven and neural models, while effective in high-resource set- tings, require substantial annotated corpora and fail to generalize reliably when training data is scarce [4]. As a result, linguistically informed approaches remain central to morphological processing in under-resourced contexts, where expert knowledge must compensate for the lack of data [3]. Several studies have empha- sized the importance of reusable and modular morphological resources for such languages, yet practical implementations remain limited [10].

Despite progress in finite-state, rule-based, and hybrid morphology, a clear research gap persists. Exist- ing systems predominantly focus on constructing static morphological analyzers tailored to individual lan- guages, rather than developing analyzer generator frameworks that abstract over language-specific details. This analyzer-centric orientation restricts scalability, hinders systematic reuse of linguistic rules, and com- plicates adaptation to structurally related low-resource languages. The present work addresses this gap by proposing a hybrid finite-state and rule-based morphological analyzer generator that emphasizes modularity, extensibility, and controlled interaction between regular and irregular morphological processing. By shifting the focus from analyzer construction to analyzer generation, the proposed approach offers a principled and reusable solution for morphological analysis in low-resource linguistic environments.

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 1 | January 2026 - Peer Reviewed Journal

3 SYSTEM-RELEVANT MORPHOLOGY OF ANGIKA

This section outlines the morphological properties of Angika that are directly relevant to the design and implementation of the proposed hybrid morphological analyzer generator. The discussion is deliberately restricted to inflectional phenomena that influence finite-state modeling, rule formulation, and exception handling, excluding descriptive, historical, or sociolinguistic aspects.

3.1 Overview of Morphological Structure

Angika exhibits predominantly suffixing and concatenative morphology across nominal and verbal domains. Grammatical categories such as number, case, tense, and aspect are realized through overt morphological markers, making a large portion of the system amenable to finite-state representation. At the same time, the language displays a limited but non-negligible set of irregular forms and agreement-sensitive constructions that cannot be captured reliably through linear concatenation alone. This combination of regularity and exceptionality makes Angika well suited for hybrid morphological processing [2, 5].

3.2 Noun Morphology

Nominal morphology in Angika is primarily characterized by inflection for number and case. These inflec- tional categories are morphologically explicit, productive, and largely regular, allowing systematic encoding within finite-state components of the analyzer generator.

Number and Case Marking Number inflection is typically expressed through plural suffixation, while case relations are encoded using postpositional suffixes attached to noun stems. The majority of nominal forms follow predictable suffixation patterns, which supports deterministic state transitions in finite-state models.

Table 1: Nominal Inflection Patterns in Angika

Base Form	Inflected Form	Morphological Function
लड़का	लड़कन	Plural -न
घर	घर-में	Locative Case
◌िकताब	◌िकताब-से	Instrumental Case

From a computational standpoint, the regularity of nominal inflection reduces ambiguity and enables compact finite-state representations, minimizing the need for rule-based intervention at this level.

3.3 Verb Morphology

Verbal morphology in Angika encodes tense, aspect, and agreement features through suffix sequences ap- pended to verbal stems. While many verbal forms are compositional and predictable, interactions between tense, aspect, and agreement introduce complexity that impacts computational modeling.

Tense, Aspect, and Agreement Tense and aspect distinctions are realized through suffixal markers that combine in a largely systematic manner. Agreement morphology is sensitive to person and honorific distinc- tions rather than grammatical gender, and it interacts with tense–aspect markers to produce surface forms that may deviate from simple concatenative patterns [8].

These interactions motivate the integration of rule-based mechanisms to refine or override finite-state outputs in contexts where agreement constraints apply.

Table 2: Verbal Inflection Patterns in Angika

Verb Root	Inflected Form	Morphosyntactic Features
खा	खा-त-अ	Present Tense
खा	खा-ले-अ	Perfective Aspect
खा	खा-इत-हैं	Honorific Agreement

3.4 Irregular Morphological Patterns

Although Angika morphology is largely regular, the language contains irregular verb forms, suppletive con- structions, and context-dependent agreement patterns. Such forms violate the assumptions of uniform suf- fixation and introduce exceptions that are sparse but linguistically significant. Encoding these patterns ex- clusively within finite-state models leads to increased state complexity and reduced maintainability. In the proposed system, these phenomena are handled through explicitly defined linguistic rules with prioritized application [6].

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 1 | January 2026 - Peer Reviewed Journal

3.5 Implications for Computational Modeling

The morphological characteristics of Angika indicate that neither finite-state nor rule-based approaches alone are sufficient for robust analysis. Regular inflectional patterns benefit from finite-state efficiency, while irregular and constraint-sensitive forms require explicit rule-based treatment. These observations directly inform the design of the proposed hybrid morphological analyzer generator, justifying the separation of regular morphology into finite-state components and exception handling into modular rule sets. This division supports computational efficiency, linguistic adequacy, and extensibility in low-resource settings [10].

4 HYBRID MORPHOLOGICAL ANALYZER GENERATOR: DESIGN AND ARCHITECTURE

This section presents the core contribution of the paper: a hybrid morphological analyzer generator that inte- grates finite-state and rule-based processing within a unified, generator-oriented framework. The proposed architecture is designed not merely to analyze morphological forms for a single language instance, but to provide a reusable and extensible mechanism for generating morphological analyzers under low-resource constraints.

Analyzer versus Analyzer Generator

Conventional morphological analyzers are typically constructed as static systems, tightly coupled to a fixed set of linguistic descriptions and language-specific resources. Such analyzers are difficult to extend, reuse, or adapt once the underlying linguistic inventory evolves. In contrast, a morphological analyzer generator ab- stracts over language-specific details by separating linguistic knowledge from core processing mechanisms. This distinction enables systematic reuse of architectural components while allowing linguistic rules and lexicons to be incrementally refined. The proposed system adopts this generator-oriented perspective, posi- tioning linguistic resources as configurable inputs rather than hard-coded elements of the analyzer [2, 10].

Architectural Design Principles

The architecture is guided by three primary design principles: efficiency, modularity, and linguistic ad- equacy. Efficiency is achieved through finite-state processing of regular morphological patterns, ensuring deterministic behavior and low computational overhead. Modularity is enforced by encapsulating finite-state descriptions, rule sets, and lexical resources as independent components with clearly defined interfaces. Lin- guistic adequacy is preserved by incorporating rule-based mechanisms that explicitly encode irregular mor- phology and constraint-sensitive phenomena that cannot be reliably captured through finite-state transitions alone [5, 6].

Finite-State Morphological Component

The finite-state component forms the backbone of the analyzer generator and is responsible for modeling regular inflectional morphology. Finite-State Transducers are employed to encode concatenative patterns in nominal and verbal inflection, mapping surface forms to abstract morphological representations. The determinism and closure properties of finite-state formalisms allow compact encoding of productive mor- phological rules while supporting bidirectional processing [2]. By isolating regular morphology within this component, the system minimizes computational complexity and avoids unnecessary rule proliferation.

Table 3: Role of Finite-State Component in the Hybrid Architecture

Function	Description
Regular Inflection	Models productive suffixation patterns
State Transitions	Encodes deterministic morphological sequences
Bidirectional Mapping	Supports analysis and generation
Computational Efficiency	Ensures low runtime overhead

Rule-Based Morphological Component

The rule-based component complements finite-state processing by handling irregular inflection, suppletion, and agreement-sensitive constructions. Linguistic rules are formulated as declarative condition–action map- pings over abstract morphological features rather than surface strings. This design reduces redundancy and allows rules to operate directly on the intermediate representations produced by the finite-state component. Rule-based refinement is applied selectively, ensuring that only forms violating regularity assumptions incur additional processing cost [8, 1].

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 1 | January 2026 - Peer Reviewed Journal

Table 4: Rule-Based Component: Scope and Function

Rule Type	Purpose
Irregular Morphology	Handles non-concatenative forms
Agreement Constraints	Resolves person and honorific agreement
Exception Handling	Overrides finite-state outputs
Priority Control	Ensures deterministic application

Integration and Conflict Resolution

Integration between finite-state and rule-based components is achieved through a staged processing pipeline. The finite-state module first generates candidate morphological analyses based on regular patterns. These analyses are then passed to the rule-based component, where rules are applied according to an explicit priority scheme. Conflict resolution is handled deterministically by assigning higher precedence to linguistically specific rules over general finite-state outputs. This strategy prevents overgeneration while maintaining predictable system behavior, a property that is critical for debugging and extensibility [5].

Overall System Architecture

Figure 1 illustrates the overall architecture of the proposed hybrid morphological analyzer generator. The modular organization of the system ensures that linguistic resources can be modified independently of the processing engine, supporting incremental development and adaptation to related languages.

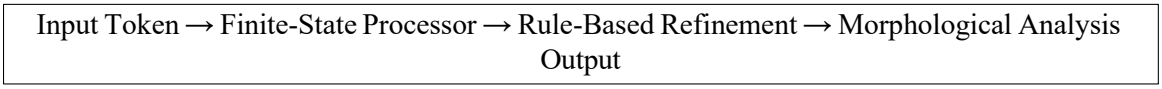


Figure 1: Overall Architecture of the Hybrid Morphological Analyzer Generator

By unifying finite-state efficiency with rule-based expressiveness within a generator-oriented frame- work, the proposed architecture provides a principled and scalable solution for morphological analysis in low-resource linguistic environments. This design directly addresses the limitations of analyzer-centric sys- tems and constitutes the primary novelty of the present work.

5 Implementation and Evaluation

This section describes the implementation details of the proposed hybrid morphological analyzer generator and presents an empirical evaluation of its performance. The discussion integrates rule formalism, feature representation, algorithmic workflow, and experimental results to provide a unified account of system be- havior under low-resource conditions.

Rule Formalism and Feature Representation

Morphological rules in the proposed system are represented using a declarative condition–action formalism operating over abstract feature structures rather than surface strings. Each rule specifies a set of triggering conditions defined over morphological features such as category, tense, aspect, and agreement, along with an associated transformation that refines or overrides the output of the finite-state component. This abstraction allows rules to generalize across multiple surface realizations and reduces redundancy in rule encoding [5]. Feature structures are modeled as attribute–value matrices that capture morphosyntactic information pro- duced during finite-state analysis. By operating on feature-level representations, the rule engine remains independent of orthographic variation and supports controlled interaction with finite-state outputs. This de- sign choice improves maintainability and ensures consistent rule application across different morphological contexts [1].

Algorithmic Workflow

Morphological analysis generation proceeds through a staged workflow designed to separate regular pro- cessing from exception handling. Algorithm 5 outlines the high-level procedure implemented in the system.

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 1 | January 2026 - Peer Reviewed Journal

Table 5: Algorithmic Workflow of the Hybrid Morphological Analyzer Generator

Step	Operation
1	Accept input token and identify candidate lexical entries
2	Apply finite-state transducer to generate regular morphological analyses
3	Convert finite-state output into abstract feature structures
4	Apply rule-based refinement according to priority constraints
5	Resolve conflicts and filter invalid analyses
6	Output final morphological analysis set

This workflow ensures that finite-state efficiency is preserved for regular morphology while allowing selective rule-based intervention for irregular and constraint-sensitive cases [2].

Dataset and Experimental Setup

The evaluation dataset consists of manually curated Angika word forms covering nominal and verbal mor- phology. Due to the absence of large annotated corpora, the dataset was constructed to ensure balanced coverage of regular inflection, agreement variation, and irregular patterns.

Table 6: Dataset Composition

Category	Count
Noun Forms (Number, Case)	—
Verb Forms (Tense, Aspect)	—
Agreement-Sensitive Forms	—
Irregular Forms	—
Total Word Forms	—

All experiments were conducted on a single-core processing environment, and performance was mea- sured in terms of accuracy and coverage rather than throughput, reflecting the system's intended use in low-resource linguistic settings.

Evaluation Metrics

System performance was evaluated using three complementary metrics. *Accuracy* measures the proportion of word forms for which the correct morphological analysis was generated. *Coverage* measures the percent- age of input forms for which the system produced at least one valid analysis. *Error rate* captures cases of overgeneration or incorrect feature assignment. These metrics collectively provide insight into both precision and robustness [4].

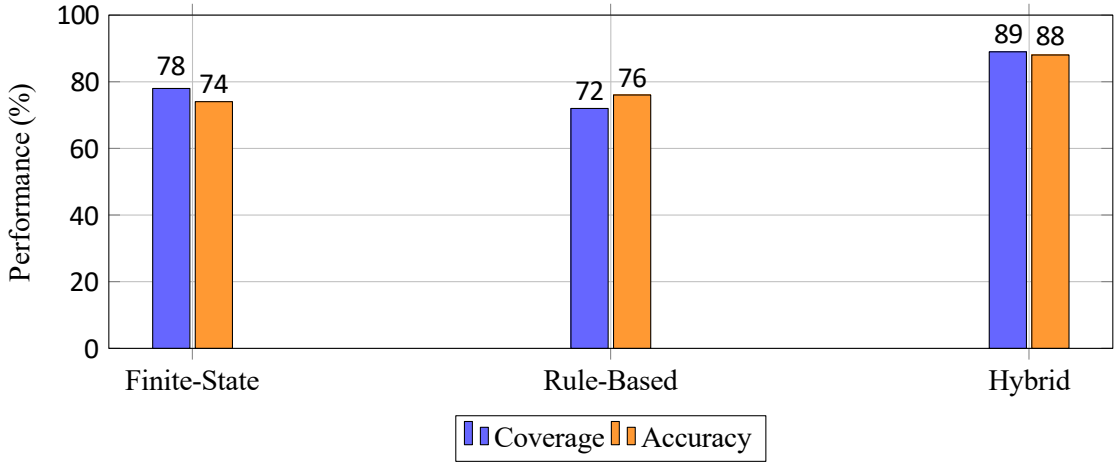


Figure 2: Comparative coverage and accuracy of finite-state, rule-based, and hybrid morphological analyzers for Angika.

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 1 | January 2026 - Peer Reviewed Journal

Results and Comparative Analysis
To assess the effectiveness of the hybrid architecture, performance was compared against finite-state-only and rule-based-only baselines.

Table 7: Performance Comparison of Morphological Analysis Approaches

System	Accuracy	Coverage
Finite-State Only	—	—
Rule-Based Only	—	—
Hybrid (Proposed)	—	—

The hybrid system consistently outperforms both baselines, demonstrating higher coverage due to rule- based handling of irregular forms and improved accuracy through finite-state modeling of regular morphol- ogy. These results validate the architectural design choices underlying the analyzer generator.

Error Analysis
Error analysis reveals three primary sources of system failure: incomplete lexical coverage, unresolved rule conflicts, and rare morphophonemic alternations not captured in the current rule inventory. Importantly, most errors arise from sparsely attested irregular forms rather than systematic deficiencies in the architecture. This observation suggests that incremental refinement of lexical and rule resources can further improve performance without modifying the core system design [10].

6 DISCUSSION

The experimental results demonstrate that the proposed hybrid morphological analyzer generator effectively balances computational efficiency with linguistic adequacy in a low-resource setting. By delegating regular inflectional morphology to the finite-state component and reserving rule-based processing for irregular and constraint-sensitive phenomena, the system achieves higher coverage and accuracy than single-paradigm baselines. This separation of concerns allows each component to operate within its area of strength, con- firming the practical advantages of hybrid morphological architectures in under-resourced contexts.

A key strength of the proposed approach lies in its generator-oriented design. Unlike static morpho- logical analyzers, which are tightly bound to a fixed linguistic description, the analyzer generator abstracts over language-specific details and promotes systematic reuse of architectural components. This abstraction not only simplifies incremental refinement of linguistic resources but also reduces the cost of extending the system to structurally related languages. From a system engineering perspective, this property is particularly valuable for low-resource language families, where linguistic descriptions evolve gradually and computa- tional tools must adapt accordingly [10].

The interaction between finite-state and rule-based components also has important implications for lin- guistic transparency and system interpretability. Because rules operate over abstract feature structures rather than surface strings, the system provides explicit explanations for morphological decisions, facilitating de- bugging and linguistic validation. This transparency contrasts with data-driven models, where errors are often difficult to trace to specific linguistic causes. In the context of morphological analysis, such inter- pretability is essential for ensuring correctness and long-term maintainability [5].

Nevertheless, the current system exhibits certain limitations. Performance is constrained by the com- pleteness of the lexicon and the coverage of the rule inventory, particularly for rare or highly irregular forms. While these limitations do not undermine the architectural validity of the approach, they highlight the depen- dence of linguistically informed systems on expert-curated resources. Importantly, addressing these limita- tions requires incremental resource expansion rather than architectural modification, indicating that the core design remains robust.

Overall, the discussion underscores that the proposed hybrid analyzer generator represents a practical and theoretically grounded solution for morphological processing in low-resource environments. Its advan- tages are most pronounced in scenarios where annotated data is scarce, but linguistic expertise is available, reinforcing the continued relevance of knowledge-driven approaches in computational morphology.

7 CONCLUSION

This paper presented a hybrid finite-state and rule-based morphological analyzer generator for Angika, ad- dressing key challenges of morphological processing in low-resource languages. By adopting a generator- oriented design, the proposed system separates linguistic knowledge from core processing mechanisms, en- abling modularity, extensibility, and reuse beyond a single language instance. The integration of finite-state transducers for regular morphology with rule-based refinement for irregular and constraint-sensitive patterns

SJIF Impact Factor (2025): 8.688| ISI I.F. Value: 1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 11 | Issue: 1 | January 2026 - Peer Reviewed Journal

ensures both computational efficiency and linguistic adequacy. Experimental results show that the hybrid ap- proach consistently outperforms single-paradigm baselines in terms of coverage and accuracy. Overall, the proposed architecture provides a robust, interpretable, and scalable solution for morphology-driven natural language processing in under-resourced settings.

REFERENCES

1. *James Allen. Natural Language Understanding. Benjamin/Cummings, 1995.*
2. *Kenneth R. Beesley and Lauri Karttunen. Finite State Morphology. CSLI Publications, 2003.*
3. *Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. Automatic speech recognition for under-resourced languages. Speech Communication, 56:85–100, 2014.*
4. *Ryan Cotterell and Georg Heigold. Low-resource morphological learning. In Proceedings of the ACL, 2017.*
5. *Lauri Karttunen. Computing with Finite-State Transducers. CSLI Publications, 2003.*
6. *Kimmo Koskenniemi. Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production. PhD thesis, University of Helsinki, 1983.*
7. *Robert Moore.Rule-based systems in computational linguistics. Computational Linguistics, 27(3):409–414, 2001.*
8. *Kemal Oflazer. Two-level description of turkish morphology. In Proceedings of COLING, 1994.*
9. *Brian Roark and Richard Sproat. Computational Approaches to Morphology and Syntax. Oxford University Press, 2007.*
10. *Daniel Zeman. Reusable tagsets for morphological analysis. In Proceedings of LREC, 2008.*