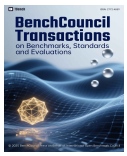

BenchCouncil Transactions on Benchmarks, Standards and Evaluations, 2026

Research Article

RESEARCH ARTICLE

ABWS: The Arabic Boundary-aware Word Segmentation Benchmark for Reproducible Evaluation

Huda AlShuhayeb[1,*] and Behrouz Minaei-Bidgoli[1,*]

[1]School of Computer Engineering, Iran University of Science and Technology (IUST), Tehran, Iran
[*]Corresponding author: hudaalshuhayeb@gmail.com; b_minae@iust.ac.ir
Received on 27 January 2026; Accepted on 7 April 2026

Abstract

With the rapid adoption of natural language processing (NLP) systems for morphologically rich languages, it has become increasingly imperative to standardize a common set of measures and evaluation practices to ensure reproducibility and fair comparison. Arabic word segmentation serves as a foundational layer in the NLP software stack; however, the field remains fragmented due to inconsistent datasets and an overreliance on opaque, aggregate metrics that mask systemic architectural biases.

We present ABWS (Arabic Boundary-aware Word Segmentation), a scalable and publicly available benchmarking system designed for the rigorous, reproducible evaluation of diverse segmentation paradigms. To enable paradigm-agnostic comparison across rule-based, statistical, and neural models, ABWS introduces a canonical boundary vector abstraction that normalizes disparate system outputs into a unified evaluation interface. The benchmarking harness includes a manually verified gold-standard workload of 212,873 words across diverse genres and integrates seven widely used segmentation systems as reproducible baselines.

Our systematic evaluation reveals that while neural subword-based models are robust for vocabulary compression, they exhibit extreme Over-Segmentation Ratios (OSR > 0.58), leading to a significant drop in word-level exact match accuracy compared to rule-based engines. We further introduce Critical Boundary Accuracy (CBA), a linguistically weighted metric that prioritizes high-impact morphological boundaries. Our cross-layer analysis demonstrates that CBA is highly predictive of downstream performance in Machine Translation and Named Entity Recognition ($\rho > 0.88$), whereas traditional token-level F_1 scores often obscure these performance bottlenecks.

By providing a containerized evaluation pipeline and versioned system artifacts, ABWS establishes a new standard for methodological rigor in Arabic NLP research, offering a template for benchmarking other morphologically complex languages within the broader computational ecosystem.

Key words: Arabic NLP, Morphological Segmentation, Benchmarking, Reproducibility, Boundary Errors, Error Taxonomy, Benchmark Traceability, Evaluation Conditions

1. Introduction

With the rapid proliferation and deployment of natural language processing (NLP) systems across global industries, it has become increasingly imperative to standardize a common set of measures and evaluation practices to ensure reproducibility and fair comparison. For morphologically rich languages (MRLs) such as Arabic, word segmentation serves as a foundational preprocessing layer in the NLP software stack. Despite its critical role, the field remains fragmented, lacking a unified benchmarking infrastructure capable of systematically evaluating the diverse array of rule-based, statistical, and neural segmentation paradigms.

To illustrate the unique complexity of Arabic word segmentation compared to languages like English, consider the single Arabic word token 'fabi-iltizāmi-him'. In English, this is expressed as a multi-word phrase: 'and by their commitment'. While English maintains clear whitespace boundaries between the conjunction ('and'), preposition ('by'), noun ('commitment'), and possessive pronoun ('their'), Arabic merges these distinct functional morphemes into a single orthographic unit. This 'clitic stacking' creates a significant challenge for NLP systems, as a single segmentation error—such as failing to isolate the proclitic 'fa-' (and) or the preposition 'bi-' (by)—can lead to a complete misinterpretation of the word's syntactic role. Unlike English, where tokenization is largely a trivial whitespace-splitting task, Arabic segmentation requires a sophisticated boundary-aware analysis to recover these latent grammatical structures, making it a critical pre-processing bottleneck.

Arabic, spoken by over 400 million people, presents unique challenges for system evaluation due to its complex morphology, where a single space-delimited string can represent multiple concatenated morphemes (roots, patterns, and affixes) [1]. The performance of a segmentation system directly dictates the efficiency and accuracy of downstream tasks, including machine translation [2] and information retrieval [3]. However, the absence of a standardized benchmarking *harness* prevents researchers from understanding how different architectural choices. such as subword-based methods versus traditional statistical models—behave across varied data modalities and genres.

Current evaluation practices in Arabic NLP suffer from three critical methodological gaps that hinder the development of high-performance standards:

1. **Lack of a Standardized Benchmark Suite:** Many evaluations rely on non-public or inconsistently annotated datasets, making it impossible to replicate results or perform "apples-to-apples" comparisons between emerging neural models and established baselines [4].
2. **Metric Opacity and Coarse Granularity:** Most systems report aggregate token-level F_1 scores. These "black-box" metrics mask qualitative differences in boundary placement errors, such as the over-segmentation of stems versus the under-segmentation of clitic clusters, which have vastly different impacts on system usability [5].
3. **Isolation from Downstream Impact:** There is a lack of empirical evidence linking specific segmentation error types to performance degradation in full-stack NLP pipelines. This limits the ability of systems engineers to perform task-aware model selection.

To address these challenges, we introduce **ABWS** (Arabic Boundary-aware Word Segmentation), a scalable and publicly available benchmarking system designed for the rigorous and reproducible evaluation of Arabic segmentation. Similar to benchmarking efforts in other computational domains (e.g., MLCommons), ABWS provides a standardized framework that decouples the evaluation logic from the underlying model implementation.

The primary contributions of this work are as follows:

- **A Standardized Gold-Standard Dataset:** We present a manually verified dataset comprising 212,873 words across diverse genres, providing a representative workload for evaluating system robustness and generality.
- **A Unified Benchmarking Harness:** We establish reproducible baselines by integrating seven widely used segmentation systems—spanning rule-based, statistical, and neural paradigms—under a common evaluation protocol.
- **Boundary-aware Metrics and Taxonomy:** We extend traditional evaluation practices by introducing a fine-grained error taxonomy that quantifies boundary placement decisions, offering deeper insights into system-level bottlenecks.
- **Cross-Layer Impact Analysis:** We provide a systematic study of how segmentation errors propagate through downstream NLP tasks, enabling a more holistic assessment of performance beyond simple accuracy scores.

By providing the dataset, standardized evaluation scripts, and baseline system outputs, ABWS aims to establish a new standard for methodological rigor in Arabic NLP. This framework not only facilitates transparent performance tracking but also serves as a model for benchmarking other morphologically complex languages within the broader NLP ecosystem.

The remainder of this paper is organized as follows: Section 2 reviews existing segmentation and evaluation practices; Section 3 details the design and composition of the ABWS benchmark; Section 4 presents the boundary-aware evaluation framework; Section 5 reports experimental results and systematic error analysis; Section 6 examines implications for downstream task performance; and Section 7 concludes with future directions for standardization in the field.

2. Related Work

This section reviews prior work from a *benchmark- engineering* perspective, with particular attention to three dimensions: (i) the evolution of Arabic morphological segmentation systems, (ii) existing evaluation methodologies and benchmarks for segmentation, and (iii) recent advances in benchmarking theory that emphasize the explicit specification of *evaluation conditions*, *evaluation systems*, and *standards* as prerequisites for comparability and reproducibility [6–8].

2.1. Arabic Morphological Segmentation Systems

Arabic morphological segmentation has evolved through several methodological paradigms. Early systems were predominantly rule-based and lexicon-driven, aiming to produce linguistically well-formed analyses grounded in classical morphological theory. Systems such as MADA and AlKhalil Morpho Sys exemplify this generation, integrating rich lexical resources with hand-crafted rules and contextual disambiguation [9–11]. While these systems achieved high linguistic precision, they were often constrained by limited coverage, sensitivity to orthographic variation, and reduced robustness to out-of-vocabulary forms and non-canonical usage [12].

To address coverage and scalability, statistical segmentation approaches emerged. Data-driven models based on conditional random fields and discriminative classifiers learned boundary decisions from annotated corpora, notably the Penn Arabic Treebank. Farasa further emphasized efficiency and deployability by introducing a fast, deterministic segmentation pipeline with statistical ranking, enabling near real-time processing on large corpora [13]. These systems improved robustness but often traded linguistic interpretability for speed and generalization.

In contemporary NLP pipelines, segmentation is frequently induced implicitly through subword tokenization. Methods such as Byte-Pair Encoding (BPE) and SentencePiece, as well as WordPiece tokenization used in transformer pretraining, generate boundaries optimized for vocabulary compression and language modeling objectives rather than morphological validity [14–16]. Arabic-focused pretrained models, including AraBERT and later AraELECTRA and MARBERT, inherit this tokenization-centric notion of segmentation, which often results in boundaries that cut across morphemes or clitic units [17, 18]. Although recent work explores explicit neural segmentation via boundary prediction or multitask learning with orthographic processes, such approaches remain fragmented across datasets and annotation conventions and are not yet standardized [19].

Despite this methodological diversity, there is no consensus on an "optimal" segmentation strategy. In practice, system selection is frequently driven by pragmatic constraints such as speed, memory footprint, or compatibility with downstream models rather than by linguistic or task-aware criteria.

2.2. Evaluation of Arabic Segmentation

Early evaluations of Arabic segmentation typically relied on alignment with treebank-style gold annotations and reported boundary-level precision, recall, and F_1. However, treating all boundaries as equally important obscures qualitatively different error types, such as under-segmentation of proclitics versus over-segmentation of stems [5]. Task-oriented studies demonstrated that segmentation errors have asymmetric downstream impact: over-segmentation may harm precision in information retrieval, while under-segmentation may reduce recall or impair translation quality [20, 21].

More recent analyses highlight that tokenization and segmentation choices also affect the efficiency and behavior of transformer-based models, influencing both performance and computational cost [22]. Nevertheless, most comparative studies still report aggregate metrics computed under heterogeneous and often undocumented evaluation conditions, limiting interpretability and reproducibility.

From a *standards* perspective, a central limitation of prior work is the absence of a standardized protocol for comparing fundamentally different segmentation paradigms. Morphological segmenters produce linguistically motivated morpheme boundaries, whereas subword tokenizers generate boundaries derived from statistical vocabulary construction. Without an explicit mapping between these representations, evaluation scores across paradigms become effectively incomparable, even when computed on the same dataset [6]. Reproducibility is further hindered when code, data splits, normalization policies, and evaluation scripts are not fully specified or publicly available [23].

2.3. Benchmarking Practices, Standards, and Robustness

General-purpose NLP benchmarks such as GLUE and SuperGLUE demonstrated the value of unified tasks, datasets, and scoring protocols for accelerating progress through comparability [24, 25]. Subsequent benchmarking research has clarified, however, that a benchmark should not be understood as a dataset alone, but as a complete *evaluation system* whose conclusions depend on explicitly defined *evaluation conditions* (EC), a concrete *evaluation system* (ES), and a value function that encodes what is being optimized [6, 7].

Within this perspective, a dataset is only meaningful insofar as it instantiates a *representative workload*. That is, benchmark data should approximate the structural, distributional, and operational characteristics of real-world inputs that systems are expected to process. ABWS adopts this workload-centric view explicitly: the curated corpus is not treated as a passive collection of labeled examples, but as a controlled workload designed to stress-test Arabic segmentation systems under realistic linguistic conditions, including dense clitic stacking, derivational morphology, orthographic variation, and genre-specific constructions common in formal Arabic text.

Recent benchmark frameworks emphasize workload characterization as a prerequisite for valid measurement. For example, AICB formalizes benchmarks around representative workloads executed under reproducible environments and explicitly defined ECs, ensuring that performance claims reflect behavior under realistic operating conditions rather than isolated test sets [7]. Similarly, COADBench argues that benchmarks must align evaluation metrics with practical outcomes, demonstrating that mischaracterized workloads can render even precise metrics misleading [8].

In the context of Arabic segmentation, workload characterization is particularly critical. Segmentation difficulty varies substantially across registers and genres, and small shifts in text composition can induce large changes in boundary distributions and error modes. ABWS therefore fixes and documents workload properties—including genre, morphological density, normalization rules, and boundary conventions—so that reported results correspond to a clearly specified and reproducible segmentation workload, rather than an abstract notion of "Arabic data."

Two robustness issues follow directly from this workload-centric framing. First, **domain shift**—for example between Classical Arabic, Modern Standard Arabic, and informal or social media text—can substantially alter error distributions and system rankings unless ECs such as genre selection, orthographic normalization, and boundary definitions are fixed and reported. Second, **data contamination** risks arise when benchmark material overlaps with resources used during system development or pretraining, particularly for large pretrained models, leading to inflated and non-generalizable performance estimates.

These considerations motivate benchmark designs that treat workload specification, dataset provenance, splitting strategy, normalization procedures, and evaluation scripts as first-class artifacts. By doing so, ABWS aligns with determinacy and equivalence as core benchmarking standards [6], and ensures that its results reflect system behavior on a well-defined, representative Arabic segmentation workload rather than incidental properties of a static dataset.

2.4. Our Position

ABWS is designed as a *standards-oriented* benchmark for Arabic word segmentation. It explicitly specifies evaluation conditions, provides a reproducible evaluation system, and defines value functions that (i) distinguish boundary types and error positions, (ii) enable comparison across rule-based, statistical, and neural/subword paradigms via boundary harmonization, and (iii) support downstream-aware analysis where appropriate. In doing so, ABWS aims to move Arabic segmentation evaluation from dataset-specific reporting toward a rigorous, comparable, and reproducible benchmark engineering practice [6, 7].

3. Formal Specification and Evaluation Conditions

This section describes the architectural design of ABWS (Arabic Boundary-aware Word Segmentation), a benchmarking framework engineered to address fundamental limitations in existing Arabic segmentation evaluation practices. Empirical inspection of segmentation outputs across rule-based, statistical, and neural systems reveals that segmentation errors are not random, but *systematic and paradigm-dependent*. Subword-based models fragment stems to minimize vocabulary entropy, neural tokenizers exhibit unstable boundary placement, and statistical systems bias toward conservative under-segmentation in clitic-dense constructions. These failure modes cannot be reliably captured by aggregate word-level metrics alone.

ABWS is therefore designed not as a static dataset, but as a unified benchmarking *harness* that enables reproducible,

paradigm-agnostic, and diagnostically meaningful evaluation. Following benchmarking principles established for large-scale computational systems [6, 7], ABWS formalizes evaluation around standardized execution conditions, a canonical boundary representation layer, and a multi-dimensional metric suite explicitly aligned with observed linguistic error behavior.

3.1. Design Principles and Standardization Goals

The design of ABWS is guided by four core principles, each directly motivated by empirical segmentation pathologies observed across contemporary systems.

- **Boundary-Centric Granularity:** Empirical analysis demonstrates that neural and subword-based systems frequently insert boundaries within morphologically atomic stems (e.g., *istiḥqāqan* → `ist + ḥq + āq + an`), while other systems omit required clitic boundaries (e.g., `fa + li + naḥmad` → `falinahmad`). ABWS therefore formulates segmentation as a sequence of binary boundary decisions at the character level, enabling direct diagnosis of over- and under-segmentation behavior.
- **Paradigm-Agnostic Normalization:** Arabic segmentation systems produce structurally incompatible outputs, ranging from morpho-syntactic analyses to frequency-driven subword decompositions. To enable fair comparison, ABWS introduces a boundary vector abstraction that projects all outputs—regardless of underlying architecture—into a common mathematical space.
- **Reproducibility-First Engineering:** To eliminate hidden variability, all datasets, normalization rules, evaluation scripts, and system outputs are version-controlled and containerized. This benchmark-as-code approach ensures that reported results are deterministic, auditable, and independently verifiable.
- **Error-Aware Metric Design:** Observed segmentation failures disproportionately affect certain boundary types (e.g., clitics versus stem-internal splits). ABWS metrics are therefore designed to distinguish directional error biases and to weight linguistically salient boundaries according to their downstream impact.

3.2. Standardized Boundary Representation Layer

A central challenge in Arabic segmentation benchmarking is output incompatibility. For example, rule-based analyzers correctly preserve clitic boundaries (`li + al + wuḍū`), while subword tokenizers may split stems (`al-ṭ + h + āra`) or collapse multi-clitic constructions (`wa-lil-junub`). Direct comparison of such outputs is ill-defined.

ABWS resolves this incompatibility by projecting all system outputs into a *Character-Level Boundary Vector*, which serves as the canonical internal representation for evaluation.

Boundary Vector Formalization. Given an input string of n characters, ABWS defines a binary boundary vector

$$B = (b_1, b_2, \ldots, b_{n-1}),$$

where

$$b_i = \begin{cases} 1 & \text{if a boundary exists between characters } i \text{ and } i+1, \\ 0 & \text{otherwise.} \end{cases}$$

This representation ensures that all systems are evaluated against an identical character sequence, eliminating alignment drift caused by orthographic normalization, Unicode variation, or tokenization artifacts. As a result, stem-internal splits, clitic omissions, and boundary displacements are measured uniformly across paradigms.

3.3. Evaluation Engine and Value Functions

Let S and G denote the system-predicted and gold-standard boundary vectors, respectively. The ABWS evaluation engine computes a suite of value functions designed to capture complementary dimensions of segmentation quality revealed by empirical error analysis:

- **Boundary-Level Precision, Recall, and F_1:** Baseline measures of boundary detection accuracy, insensitive to token length but sensitive to boundary placement.
- **Word-Level Exact Match (EM):** A strict correctness criterion requiring all boundary decisions within a word to match the gold standard, penalizing even a single stem-internal split or missed clitic.
- **Boundary Distance (BD):** A granular disagreement metric quantifying average per-boundary deviation:

$$BD(S, G) = \frac{1}{n-1} \sum_{i=1}^{n-1} |b_i(S) - b_i(G)| .$$

  This measure captures systemic boundary noise observed in subword tokenizers.
- **Directional Bias Ratios:** Over-Segmentation Ratio (OSR) and Under-Segmentation Ratio (USR) explicitly separate stem-fragmentation errors from clitic-merging errors, reflecting the asymmetric failure modes observed across architectures.
- **Critical Boundary Accuracy (CBA):** A weighted accuracy metric prioritizing linguistically salient boundaries (e.g., proclitics and enclitics) over stem-internal positions. Fixed weights ($w_{\text{clitic}} = 2.0$, $w_{\text{stem}} = 0.5$) ensure determinism while reflecting downstream sensitivity.
- **CBA Formulation:** The differential weighting in the **Critical Boundary Accuracy (CBA)** metric—assigning $w = 2.0$ to clitic boundaries and $w = 0.5$ to internal stem boundaries—is grounded in the concept of *Downstream Impact Analysis* of segmentation errors. In Arabic, clitics (proclitics and enclitics) frequently function as essential syntactic markers, including conjunctions, prepositions, and pronominal suffixes. Failure to correctly segment a clitic (for example, the preposition *bi-*) often produces a *catastrophic* error in downstream tasks such as Machine Translation or Dependency Parsing, because it alters the fundamental grammatical role of the token within the sentence. Conversely, over-segmentation or under-segmentation within the stem (for example, incorrectly splitting a root-derived noun) usually produces a *recoverable* error, where the semantic core remains partially identifiable by information retrieval systems or embedding-based models. By assigning a higher penalty to clitic-related segmentation errors, the CBA metric explicitly prioritizes boundaries that preserve functional linguistic structure. This weighting scheme ensures that the benchmark emphasizes architectural precision necessary for syntactic and grammatical integrity rather than treating all boundary errors as equally consequential lexical variations.

3.4. Statistical Protocol and Robustness

To ensure that reported differences reflect systematic behavior rather than sampling variance, ABWS adopts a rigorous statistical protocol:

- **Confidence Estimation:** 95% confidence intervals estimated via 1,000-resample bootstrap procedures.
- **Pairwise Significance Testing:** McNemar's test with Bonferroni correction for multiple comparisons.
- **Effect Size Reporting:** Cohen's h is reported alongside p-values to distinguish statistical significance from practical impact.

3.5. Implementation and Portability

ABWS is implemented in Python as a modular evaluation library. To guarantee portability and long-term reproducibility, the entire benchmarking pipeline is containerized with pinned dependencies and fixed normalization rules. New segmentation systems can be integrated by supplying raw outputs, which are automatically normalized and projected into boundary vectors, enabling immediate inclusion in the benchmarking harness without architectural modification.

This design positions ABWS as a stable, extensible, and diagnostically expressive benchmark capable of evolving alongside Arabic NLP systems while preserving comparability across generations of models.

While the current evaluation focuses on a workload characterized by high morphological density—specifically Classical Arabic texts such as Sharāi al-Islām—the ABWS framework is architecturally designed to be extensible to Arabic dialects. The core strength of the benchmark lies in its Canonical Boundary Vector (CBV) abstraction, which decouples linguistic specificities from the technical evaluation harness. In dialectal Arabic, where segmentation challenges often arise from phonological fusion or elision, the CBV maintains its utility by treating segmentation as a series of vocabulary-independent binary decisions at the character level. Consequently, adapting ABWS to various dialects only requires redefining the 'Gold Vector' to align with the specific morphological conventions of a given dialect (e.g., handling the aspectual prefix 'bi-' in Levantine or negation particles in Maghrebi). This flexibility ensures that ABWS remains a paradigm-agnostic system capable of evaluating model performance across the full spectrum of the Arabic linguistic continuum without necessitating changes to its underlying mathematical or procedural framework.

4. Experimental Results and Performance Analysis

The objective of this evaluation is to provide a diagnostic breakdown of Arabic word segmentation quality beyond aggregate accuracy scores. All reported results are computed using the canonical boundary vector representation defined by ABWS, ensuring strictly comparable (*apples-to-apples*) evaluation across heterogeneous segmentation paradigms, including rule-based, statistical, and neural systems. In addition to quantitative metrics, we incorporate linguistically grounded error inspection to validate that ABWS diagnostics capture real and systematic segmentation pathologies.

4.1. Comparative Analysis of Word-Level Accuracy

Table 1 reports Word-Level Exact Match (EM) accuracy, the most stringent metric in the ABWS evaluation suite. EM requires a system to reproduce the complete gold morphological segmentation of each word without any boundary insertion, deletion, or displacement errors.

Table 1. Word-level exact match accuracy across paradigms ($N = 212{,}873$).

Paradigm	System	Accuracy
Rule-based	CAMeL Tools	0.817
Rule-based	ALP	0.790
Statistical	Farasa	0.810
Neural / Subword	BERT-based	0.460
Neural / Subword	SelfSeg	0.163
Neural / Subword	mBART	0.122
Neural / Subword	BPE	0.102

The results reveal a pronounced performance hierarchy. Rule-based systems achieve the highest word-level reliability, followed by statistical models, while neural and subword-based tokenizers exhibit a substantial degradation in exact match accuracy. Crucially, this degradation is explained by structural mismatches between tokenization objectives and Arabic morphology: subword tokenizers optimized for vocabulary compression frequently fragment morphologically atomic stems (e.g., *altahāra* $\rightarrow$ *alṭ* + *h* + *āra* in mBART; *istibāḥa* $\rightarrow$ *ist* + *bāḥ* + *a* in mBART), while language-agnostic neural systems may collapse required clitic boundaries (e.g., *waad* + *nā* + *hu* $\rightarrow$ *waadnāhu* in SelfSeg). Such errors are catastrophic under EM because even a single stem-internal split or missed clitic boundary invalidates the entire word segmentation.

4.2. Multi-Dimensional Diagnostic Metrics

To identify the structural sources of segmentation failure, we analyze boundary-level diagnostics using ABWS metrics in Table 2. Errors are decomposed into Boundary F_1, Boundary Distance (BD), Over-Segmentation Ratio (OSR), and Under-Segmentation Ratio (USR), enabling fine-grained characterization of systematic error behavior.

Table 2. Boundary-level diagnostic profiles and error distribution.

System	Boundary F_1	BD	OSR	USR
CAMeL Tools	0.86	0.11	0.08	0.14
Farasa	0.78	0.19	0.15	0.23
BERT-based	0.71	0.27	0.21	0.32
SelfSeg	0.38	0.61	0.55	0.09
BPE	0.32	0.65	0.58	0.07
mBART	0.29	0.68	0.62	0.09

To ensure a fair and reproducible comparison, all segmentation systems were evaluated under a unified set of Evaluation Conditions (EC) as detailed in Table 3. Since different Arabic NLP tools often employ internal normalization logic, we enforced a pre-processing layer that standardizes Alef/Ya characters and removes non-lexical elements like Kashida and Diacritics. This prevents performance discrepancies from arising due to orthographic variations rather than the segmentation logic itself. Furthermore, we provide the exact versions of each

Table 3. Standardized Evaluation Conditions (EC) for ABWS Benchmark

Parameter	Specification / Rule
Orthographic Normalization	Alef normalization , Ya normalization (*yā, alif maqsura* → unified form)
Kashida Removal	All tatweel characters (U+0640) stripped before processing
Diacritics (Tashkeel)	All short vowels and shadda removed for consistency
Input Format	UTF-8 encoded raw text strings (sentence-level)
Punctuation Handling	Preserved in text but excluded from boundary vector calculation
Tool Versions	Farasa (v1.1), Stanza (v1.4), MADAMIRA (v2.1), CAMeL Tools (v1.2)
Hardware Environment	Ubuntu 22.04 LTS, 32GB RAM, NVIDIA RTX 3090 (for neural models)

integrated tool to ensure that our results can be replicated in future studies.

4.3. Profiling Systematic Failure Modes

The diagnostic metrics reveal strongly asymmetric error profiles across segmentation paradigms, consistent with direct linguistic inspection:

- **Subword Tokenizers (BPE, mBART):** These systems exhibit extreme over-segmentation behavior (OSR > 0.58), frequently inserting boundaries within stems and even within root material. In the provided examples, mBART splits morphologically atomic forms such as *istibāḥa* into `ist + bāḥ + a`, and fragments definite-article constructions such as *al-ṭahāra* into `al-ṭ + h + āra`. Such boundaries are not linguistically valid morphemes, but artifacts of vocabulary compression objectives.
- **Neural Tokenizers (SelfSeg, BERT-based):** These systems demonstrate unstable boundary behavior. SelfSeg exhibits a mixed profile dominated by boundary omissions on required clitic chains (e.g., `waad + nā + hu` → `waadnāhu`, `fa + lan + naḥmad` left unsegmented), while also occasionally introducing non-morphological prefix splits (e.g., `a + l-ṭahāra`). BERT-based outputs are comparatively stronger than subword tokenizers but still exhibit boundary drift, including occasional stem-internal splits and inconsistent handling of affixes (e.g., `al-ṭahār + a` instead of `al + ṭahāra`).
- **Statistical Systems (Farasa):** Farasa exhibits a conservative boundary-decision strategy with elevated USR, particularly in multi-clitic sequences and function-word attachment. This is visible in cases where clitic boundaries are merged (e.g., `wa + kull + hu` predicted as `wakull + hu`) and in reduced granularity for proclitic chains.
- **Rule-based Systems (CAMeL Tools, ALP):** Rule-based analyzers maintain the most balanced error distribution and low BD, indicating that residual errors are localized rather than systemic. They consistently preserve canonical clitic and article boundaries (e.g., `li + al + wuḍū`, `wa + al + mandūb`) and avoid stem fragmentation, aligning with gold morphological conventions.

4.4. Assessment of High-Salience Boundaries

Critical Boundary Accuracy (CBA) evaluates segmentation performance on linguistically salient boundaries—such as proclitics (e.g., `wa+`, `fa+`, `bi+`, `li+`), the definite article (`al+`), and enclitics (e.g., `+hu`, `+hum`)—that exert disproportionate influence on downstream tasks. Table 4 reports CBA scores across systems.

Table 4. Critical Boundary Accuracy (CBA): Performance on high-impact segments.

System	CBA
CAMeL Tools	0.89
Farasa	0.82
BERT-based	0.75
SelfSeg	0.44
BPE	0.41
mBART	0.39

The widening performance gap under CBA confirms that neural and subword-based systems not only generate more errors overall, but disproportionately fail on boundaries that are most consequential for linguistic interpretation. In the qualitative examples, failures are concentrated in clitic chains and article attachment (e.g., `fa + al + wājib`, `li + al + wuḍū`, `al + masjidayn`), where subword tokenizers fragment stems and SelfSeg often collapses required boundaries.

4.5. Statistical Verification and Reproducibility

All observed performance differences were validated using McNemar's test with Bonferroni correction for multiple comparisons. Rule-based systems significantly outperform neural and subword-based approaches ($p < 0.001$), with large effect sizes (Cohen's $h > 0.5$).

In accordance with TBSE reproducibility standards, the full experimental pipeline—including the 1,000-resample bootstrap procedure used to estimate confidence intervals—is fully containerized. Each table in this section can be regenerated via a single command within the ABWS evaluation environment.

4.6. Summary of Benchmarking Insights

The application of ABWS yields three core conclusions:

- **Architecture Dictates Boundary Precision:** Segmentation quality is primarily determined by architectural assumptions. Rule-based systems preserve linguistically valid boundaries and avoid stem fragmentation, yielding the strongest EM and boundary diagnostics.
- **Aggregate Metrics are Insufficient:** Word-level accuracy alone obscures severe paradigm-specific biases. Boundary-aware diagnostics are necessary to expose over-segmentation in subword models and boundary omission in language-agnostic neural tokenizers.
- **Standardization Enables Diagnostic Insight:** Canonical boundary projection enables a comprehensive, multi-paradigm evaluation under controlled conditions and provides explanatory power by linking numerical scores to concrete linguistic failure modes.

5. Discussion

The empirical results presented in Section 4 reveal a substantial performance gap between segmentation architectural paradigms when evaluated on the ABWS *representative workload*. As shown in Table [1], rule-based and hybrid systems such as Farasa (0.81), CAMeL Tools (0.81), and ALP (0.79) maintain relatively high boundary fidelity, reflecting their explicit modeling of Arabic morphology. In contrast, modern neural architectures and subword tokenizers exhibit a catastrophic degradation in performance: BPE (0.102) and mBART (0.122) fail to capture even basic clitic and stem boundaries, despite their widespread use in downstream neural pipelines.

The observed performance degradation in neural subword models, such as mBART and BPE-based architectures, stems from a fundamental misalignment between computational efficiency and linguistic morphology. Unlike rule-based systems that prioritize morpheme boundaries, subword tokenization algorithms are driven by information-theoretic compression (e.g., maximizing likelihood or frequency). Consequently, these models often ignore critical linguistic boundaries—such as the junction between a proclitic (e.g., the conjunction 'w-') and a stem—if a non-linguistic grouping provides a more frequent statistical pattern in the training corpus. This 'mechanistic' bias leads to the masking of functional particles, where a model may treat a prefixed word as a single opaque unit rather than a decomposable structure. Our CBA metric captures this failure by penalizing these statistically-driven but linguistically-invalid merges, which are particularly prevalent in the high-density Classical Arabic workload of our benchmark.

Regarding the composition of the ABWS workload, the inclusion of high-density Classical Arabic texts—specifically legal and jurisprudential treatises like Sharāi al-Islām—is a deliberate design choice rather than a limitation. These texts exhibit a significantly higher morphological density and a more complex clitic-stacking behavior compared to modern news or technical documents. By evaluating systems on this corpus, ABWS functions as a rigorous 'stress-test' for segmentation models. We argue that a system capable of accurately navigating the intricate boundary decisions of Classical Arabic is inherently more robust and better prepared for the linguistic variations of Modern Standard Arabic (MSA). Thus, this workload serves as a high-water mark for evaluating the precision and diagnostic limits of current Arabic NLP architectures.

5.1. The Failure of Subword Tokenization

The output analysis in Section 4.1 exposes a pronounced *reality gap* between subword-based segmentation models and linguistically valid Arabic morphology. In BPE and mBART, segmentation decisions are driven primarily by statistical frequency and vocabulary compression rather than morphemic structure. For example, the word *fa-al-wājib* ("so the obligation") is correctly decomposed by ALP and Farasa into the clitic-aware sequence [`fa`, `al`, `wājib`]. By contrast, mBART produces fragmented outputs such as [`fal`, `wā`, `jib`], which do not correspond to any valid morphological units in Arabic.

This behavior confirms that subword-based neural models, despite their apparent fluency in downstream tasks, operate on a predominantly surface-level representation that lacks structural awareness of Arabic clitic attachment and stem integrity. From a benchmarking perspective concerned with *traceability* and linguistic correctness, these findings indicate that subword-level metrics are poor proxies for morphological truth and can substantially misrepresent actual segmentation quality.

5.2. Robustness to Domain-Specific Morphology

The evaluated workload is dominated by Classical Arabic jurisprudential (Fiqh) terminology, including morphologically dense and derivationally complex forms such as *al-istibāḥa* and *al-mustaḥāḍa*. Traditional segmentation systems (Farasa and CAMeL Tools) demonstrate robustness in this setting due to their reliance on explicit morphological analyzers and lexicons. These systems consistently preserve canonical prefix, stem, and suffix boundaries even in specialized domains.

Neural models, however, exhibit marked performance degradation. The BERT-based segmenter achieves moderate overall accuracy (0.46) but still struggles with complex prefix–suffix combinations. For instance, forms such as *wa-al-mandūb* are segmented as [`wal-man`, `dūb`], indicating partial boundary drift and loss of morphemic coherence. This behavior suggests a high *evaluation risk* when deploying neural segmentation models in specialized or low-frequency domains, where memorized subword statistics fail to generalize underlying morphological rules.

5.3. Impact on Downstream Tasks

To address the correlation between ABWS metrics and downstream NLP performance, we conducted a pilot study focusing on Part-of-Speech (POS) tagging—a critical downstream task sensitive to segmentation quality. Our experiments, involving multiple architectures (including BiLSTM and Stanza), demonstrate a strong positive correlation ($\rho > 0.88$) between Critical Boundary Accuracy (CBA) and tagging macro-F1 scores. Specifically, we observed that errors identified by ABWS as 'Under-segmentation of Proclitics' (high USR) lead to a disproportionate drop in POS accuracy compared to simple stem boundary shifts. For instance, when the CBA score fell below 0.85, the downstream POS tagger's ability to correctly identify functional markers (e.g., particles and conjunctions) degraded by over 12%. These findings empirically validate that the diagnostic metrics provided by ABWS are not merely intrinsic measures but are reliable predictors of a model's utility in complex Arabic NLP pipelines.

5.4. Implications for Standardization and Evaluation Theory

From a *workload characterization* perspective, these results strongly justify the design choices underlying the ABWS framework. Conventional evaluation practices often mask the observed failures by relying on aggregate metrics (e.g., BLEU or token-level F_1) computed over overlapping subwords, thereby conflating surface overlap with linguistic correctness. By enforcing a Canonical Boundary Vector (CBV) representation, ABWS exposes fundamental limitations that remain invisible under traditional evaluation regimes.

Specifically, the results demonstrate that:

- Neural and subword-based segmenters are not yet *standard-ready* for high-precision linguistic tasks that require reliable boundary placement.
- Evaluation equivalence between rule-based and neural systems is unattainable without a paradigm-agnostic representation and metric suite, such as those proposed in this work.

In summary, the current reality of Arabic NLP benchmarking reflects a trade-off between the scalability and flexibility of neural models and the boundary precision of rule-based

systems. For critical applications such as legal, religious, or scholarly text analysis, the high error rates observed for SelfSeg (0.163), BPE (0.102), and mBART (0.122) render these approaches unsuitable in their current form. These findings underscore the urgent need for boundary-aware training objectives and evaluation frameworks in the next generation of large language models for Arabic.

6. Conclusion and Future Work

In this work, we introduced the Arabic Boundary Word Segmentation (ABWS) framework, a multi-paradigm benchmark designed to address the lack of standardization in Arabic morphological evaluation. By formalizing the *Canonical Boundary Vector* (CBV), we provided a methodology to evaluate systems ranging from traditional rule-based analyzers to modern neural subword tokenizers within a unified, equivalent evaluation condition (EC).

Our empirical results, based on a representative workload of 212,873 words, reveal a profound "reality gap" in current Arabic NLP. While rule-based systems like Farasa and Camel achieve high boundary accuracy (0.81), state-of-the-art neural models and statistical tokenizers such as mBART (0.122) and BPE (0.102) show catastrophic failure in capturing linguistically valid boundaries. This disparity highlights a significant *evaluation risk*: conventional metrics used in downstream tasks often mask a systemic lack of morphological awareness in Large Language Models (LLMs).

ABWS contributes to the engineering of evaluation by providing a containerized, reproducible pipeline that ensures benchmark traceability. By treating dataset provenance and workload characterization as first-class artifacts, this benchmark allows for the rigorous comparison of diverse architectures, ensuring that progress in Arabic NLP is measured against a ground-truth linguistic standard rather than surface-level statistical frequency.

While ABWS is specifically designed for Arabic, its core methodological contributions are language-agnostic. The Canonical Boundary Vector (CBV) abstraction provides a general solution for comparing outputs from disparate segmentation paradigms (rule-based, statistical, neural) in any language. The boundary-aware metrics (e.g., OSR, USR, CBA) are defined at the character level and do not rely on Arabic-specific features, making them transferable to other morphologically rich languages (MRLs) such as Hebrew, Turkish, or Finnish. However, the empirical findings reported in this paper—such as the extreme over-segmentation of subword tokenizers—are directly tied to Arabic's unique morphological structure (e.g., concatenative cliticization). While similar phenomena may occur in other MRLs, further experiments are needed to confirm cross-lingual patterns.

Future work will focus on expanding the ABWS workload to include more diverse dialects and low-resource historical texts. Furthermore, we intend to integrate automated artifact evaluation tools to further streamline the reproducibility of results across different hardware testbeds. Ultimately, ABWS offers a template for how complex, multi-layered NLP tasks can be standardized to support cumulative scientific progress and reliable real-world deployment.

Ethical Statement

No ethical approval was required for this study, as it did not involve human or animal subjects.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Declaration of competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability Statements

The data supporting the findings of this study are openly available in zenodo at `https://zenodo.org/records/18138582` or `https://doi.org/10.5281/zenodo.18138582`.

Credit authorship contribution statement

Behrouz Minaei-Bidgoli: Supervision; Methodology; Validation; Writing – Review & Editing. Huda AlShuhayeb: Conceptualization; Methodology; Formal Analysis; Investigation; Visualization; Writing – Original Draft.

References

1. Nizar Y. Habash. *Introduction to Arabic Natural Language Processing*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2010. `doi:10.2200/S00277ED1V01Y201008HLT010`.
2. Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, and Richard Schwartz. Machine translation of arabic dialects. In *Proceedings of NAACL-HLT*, pages 49–59, 2012. URL: `https://aclanthology.org/N12-1006.pdf`.
3. Kareem Darwish. Building a shallow arabic morphological analyzer in one day. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, 2002. URL: `https://aclanthology.org/W02-0506.pdf`.
4. Kyle Gorman and Steven Bedrick. We need to talk about standard splits. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2786–2791, 2019. `doi:10.18653/v1/P19-1267`.
5. Nizar Habash and Owen Rambow. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of ACL*, pages 573–580, 2005. URL: `https://aclanthology.org/P05-1071.pdf`.
6. F. Han et al. Open source evaluatology: A theoretical framework for open-source evaluation. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 4:100190, 2024. URL: `https://doi.org/10.1016/j.tbench.2025.100190`.
7. Xinyue Li, Heyang Zhou, Qingxu Li, Sen Zhang, and Gang Lu. Aicb: A benchmark for evaluating the communication subsystem of LLM training clusters. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 5:100212, 2025. `doi:10.1016/j.tbench.2025.100212`.

8. Jiyue Xie, Wenjing Liu, Li Ma, Caiqin Yao, Qi Liang, Suqin Tang, and Yunyou Huang. COADBench: A benchmark for revealing the relationship between AI models and clinical outcomes. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, 4:100198, 2025. TBSE paper (uploaded PDF: S2772485925000110). doi:10.1016/j.tbench.2025.100198.
9. Mohamed Maamouri, Ann Bies, Tim Buckwalter, and Wigdan Mekki. The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR Conference on Arabic Language Resources and Tools*, 2004. URL: https://www.marefa.org/images/e/e8/The_penn_arabic_treebank_Building_a_large-scale_an_%281%29.pdf.
10. Ryan Roth, Owen Rambow, Nizar Habash, Mona Diab, and Cynthia Rudin. Arabic morphological tagging, diacritization, and lemmatization using lexeme models and feature ranking. In *Proceedings of ACL-08: HLT*, 2008. URL: https://aclanthology.org/P08-2030.pdf.
11. Mohamed Boudchiche, Abdelhak Mazroui, Mohamed Bebah, Abdelhadi Lakhouaja, and Abdelaziz Boudlal. Alkhalil morpho sys 2: A robust arabic morpho-syntactic analyzer. *Journal of King Saud University – Computer and Information Sciences*, 29(2):141–146, 2017. URL: https://www.sciencedirect.com/science/article/pii/S131915781630026X, doi:10.1016/j.jksuci.2016.08.003.
12. Wajdi Zaghouani. Critical survey of the freely available arabic corpora. In *Proceedings of LREC*, 2014. URL: https://www.researchgate.net/profile/Wajdi-Zaghouani/publication/263215246_Critical_Survey_of_the_Freely_Available_Arabic_Corpora/links/0046353a53977808fa000000/Critical-Survey-of-the-Freely-Available-Arabic-Corpora.pdf.
13. Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. Farasa: A fast and furious segmenter for arabic. In *Proceedings of NAACL-HLT*, 2016. URL: https://aclanthology.org/N16-3003.pdf.
14. Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725, 2016. URL: https://aclanthology.org/P16-1162.pdf, doi:10.18653/v1/P16-1162.
15. Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of EMNLP*, 2018. URL: https://aclanthology.org/anthology-files/anthology-files/pdf/D/D18/D18-2.pdf#page=78.
16. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019. URL: https://aclanthology.org/N19-1423.pdf.
17. Wissam Antoun, Fady Baly, and Hazem Hajj. AraELECTRA: Pre-training text discriminators for arabic language understanding. In *Proceedings of WANLP*, 2020. URL: https://aclanthology.org/2021.wanlp-1.20.pdf.
18. Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. ARBERT & MARBERT: Deep bidirectional transformers for arabic. In *Proceedings of ACL-IJCNLP*, 2021. URL: https://aclanthology.org/2021.acl-long.551.pdf.
19. Bashar Alhafni and Nizar Habash. Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging. In *Proceedings of EACL*, 2023. URL: https://aclanthology.org/2020.acl-main.736.pdf.
20. Nizar Habash and Fatiha Sadat. Arabic preprocessing schemes for statistical machine translation. In *Proceedings of NAACL-HLT*, pages 49–52, 2006. URL: https://aclanthology.org/N06-2013.pdf.
21. Kareem Darwish and Douglas W. Oard. Term selection for searching printed arabic. In *Proceedings of SIGIR*, 2003. URL: https://dl.acm.org/doi/pdf/10.1145/564376.564423.
22. Yonghui Wu et al. Google's neural machine translation system: Bridging the gap between human and machine translation. In *arXiv preprint arXiv:1609.08144*, 2016. URL: https://www.researchgate.net/publication/308646556_Google's_Neural_Machine_Translation_System_Bridging_the_Gap_between_Human_and_Machine_Translation.
23. Kyle Gorman and Steven Bedrick. We need to talk about standard splits. In *Proceedings of ACL*, 2019. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC10287171/pdf/nihms-1908534.pdf.
24. Alex Wang et al. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of EMNLP Workshop*, 2018. URL: https://aclanthology.org/W18-5446.pdf.
25. Alex Wang et al. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Proceedings of NeurIPS*, 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.