Sociology & Cultural Research Review (SCRR)
Available Online: https://scrrjournal.com
Print ISSN: 3007-3103 Online ISSN: 3007-3111
Platform & Workflow by: Open Journal Systems
https://doi.org/10.5281/zenodo.19788013

A Unified Stemming Framework for Arabic-Script South Asian Languages: Sindhi, Urdu, and Persian

Mohsin Raza Shah
Assistant Professor, Computer Science, College Education Department, Government of Sindh
razamohsinsyed31@gmail.com
Amjad Ali Mahesar
Lecturer, College Education Department, Government of Sindh
amjad.a.mahesar@gmail.com

ABSTRACT
Stemming is a fundamental preprocessing operation in Natural Language Processing (NLP) and Information Retrieval (IR) systems, enabling the reduction of morphologically inflected or derived words to their root or stem form. Sindhi, Urdu, and Persian are three widely spoken languages that share the Perso-Arabic script, exhibit overlapping morphological structures, and carry significant lexical borrowing from one another (Ali, Khalid & Saleemi, 2019; Shah, 2016). Despite this shared linguistic substrate, all existing stemmers for these languages have been developed independently and, in a language, -specific manner, leading to redundant eU'ort, incompatible resources, and limited cross-lingual applicability. This paper proposes UPASS the Unified Perso-Arabic Script Stemmer a modular, language independent framework for stemming Sindhi, Urdu, and Persian using a shared rule architecture, a cross-lingual morpheme repository, and a unified algorithmic pipeline. The framework is built upon a detailed comparative morphological analysis of the three languages, the rule-based stripping approach validated for Sindhi secondary words (Shah, 2016), infix-stemming advances for Urdu (Ali et al., 2019), and multi-phase suU'ix-prefix removal techniques for Persian (Estahbanati & Javidan, 2009). Experimental evaluation on standard corpora demonstrates that UPASS achieves a cumulative average Stemmed Error Rate (SER) of 11.28% and an average accuracy of 88.72% across the three languages, consistently outperforming language-specific baselines.

Keywords: *Stemming, Information Retrieval, Natural Language Processing, Sindhi, Urdu, Persian, Perso-Arabic Script, Morphological Analysis, Cross-lingual NLP, Rule-Based Approach*

Introduction

The rapid growth of multilingual digital content has made it imperative for Natural Language Processing (NLP) research to extend beyond dominant languages such as English and to address the computational needs of regional and low-resource languages (Shah, 2016). South Asia is home to hundreds of languages, several of which share the Perso-Arabic script as their primary writing system. Among these, **Sindhi**, **Urdu**, and **Persian** represent a particularly compelling cluster: they share a writing script, a high degree of lexical overlap due to centuries of contact-induced borrowing, and remarkably similar morphological structures (Shah, 2016; Ali et al., 2019).

Stemming the process of reducing a word to its root or stem by removing affixes is widely recognized as the backbone of any Information Retrieval (IR) system (Shah, 2016). As Shah (2016,

p. 7) states, "stemmer is the tool which information retrieval system uses to decrease morphological variants accompanied along with the words to its stem or root." It reduces index size, improves recall, and enables effective query-document matching based on root word forms rather than exact surface forms (Shah, 2016). Despite the obvious potential for shared resources, every stemmer developed for Sindhi, Urdu, and Persian has been engineered in isolation: Assas-Band and the Light Weight Stemmer for Urdu (Akram, Naseer & Hussain, 2009), Bon and the Farsi Multi-Phase Algorithm for Persian (Estahbanati & Javidan, 2009; Khosravi, 2008), and the rule-based stripping stemmer for Sindhi (Shah, 2016). None of these systems exploits the shared morphological structure that these languages collectively possess.

This paper bridges that gap. We propose **UPASS** (Unified Perso-Arabic Script Stemmer), a modular framework that:

- Performs a systematic comparative morphological analysis of Sindhi, Urdu, and Persian
- Identifies shared prefixes, suffixes, infixes, and compound-word patterns across all three languages
- Designs a common algorithmic pipeline with language-specific extension modules
- Evaluates the unified approach against each language's standalone stemmer

The contributions of this work are fourfold:

1. The **first comparative morphological analysis** of Sindhi, Urdu, and Persian focused specifically on stemming
2. A **shared cross-lingual morpheme repository** containing 127 common prefix, suffix, and infix morphemes
3. The **UPASS algorithm**, a unified stemming pipeline applicable to all three languages with plug-in language modules
4. A **benchmarking study** demonstrating competitive accuracy against language-specific approaches

Background and Linguistic Overview

The Perso-Arabic Script Family

The Perso-Arabic script is the second most widely used writing system in the world (Al-Sughaiyer & Al-Kharashi, 2004). It is written right-to-left and is an abjad a consonant-based script where short vowels are typically omitted in everyday text. The script was originally developed for Arabic but was adapted by Persian, which introduced additional letters to represent sounds absent in Arabic (Shah, 2016). Urdu further extended the Persian alphabet, and Sindhi extended it even further, adding 18 unique letters to represent the sounds of the Indus Valley phonological system (Motlani, Tyers & Sharma, 2016).

All three languages thus share a large subset of the Unicode Perso-Arabic block (U+0600–U+06FF), creating both an opportunity for resource sharing and a challenge in the form of character homophones the same visual character may map to different Unicode code points depending on language context (Motlani et al., 2016).

Sindhi Language

Sindhi is an Indo-Aryan language spoken by approximately **44.8 million people** worldwide, including roughly 39.8 million in Pakistan and 4.98 million in India (Shah, 2016). It is the third most spoken language in Pakistan and holds official status in both Pakistan and India (Shah, 2016). Sindhi uses the Perso-Arabic script as its primary orthography in Pakistan and has **52 letters** in its alphabet (Sodhar, Sulaiman & Buller, 2023).

The language has a rich inflectional morphology: nouns inflect for gender (masculine/feminine), number (singular/plural), and case, while verbs inflect for tense, aspect, mood, person, number, and gender (Motlani et al., 2016). Sindhi words fall into two primary types — *primary* words (indivisible root words) and *secondary* words, which are further divided into *complex* words (prefix + stem + suffix) and *compound* words (two or more primary words combined) (Shah, 2016). The first publicly available morphological analyser for Sindhi was published as recently as 2016, achieving 81% naïve corpus coverage (Motlani et al., 2016). A 2023 study further confirmed that "both derivational and inflectional morphological constructions are possible in Sindhi" (Sodhar et al., 2023, p. 2900), making morphological analysis critical for all NLP applications.

Urdu Language

Urdu is spoken by approximately **200 million people** worldwide and serves as the national language of Pakistan and an official language of several Indian states (Ali et al., 2019). Its vocabulary is an eclectic blend of words from **Arabic, Persian, Turkish, Hindi, and English**, giving it a morphologically rich and complex structure (Ali et al., 2019). A distinguishing feature of Urdu is its significant **infix morphology**, inherited from Arabic patterns where morphemes are inserted within a word root rather than appended to its edges (Ali et al., 2019). Ali et al. (2019, p. 138) observe that "Urdu is robust in both inflectional and derivational morphology" and that "morpheme is the major element of Urdu morphology." This makes Urdu considerably more challenging for simple affix-stripping stemmers than languages with purely concatenative morphology.

Persian Language

Persian, also known as Farsi, is an Indo-European language of the Iranian branch spoken by over 80 million people primarily in Iran, Afghanistan, and Tajikistan. Persian morphology is primarily **agglutinative**, with a clear system of prefixes and suffixes attached to verb stems and noun roots (Estahbanati & Javidan, 2009). The first Persian stemmer, Bon (Tashakori), was built using an iterative affix-stripping algorithm and improved recall by 40% over unstemmed baselines (Ali et al., 2019). A subsequent stemmer by Mokhtaripour generated stems of Persian text **without using a language dictionary**, improving query system performance by 46% (Ali et al., 2019). Studies on Persian IR have confirmed that "light suffix-stripping procedures yield meaningful improvements in retrieval effectiveness" for Persian text (Dolamic & Savoy, 2009). Persian has fewer morphological irregularities compared to Arabic, making rule-based stemming more tractable (Khosravi, 2008).

Literature Review

Arabic Stemming

Arabic morphology is among the most complex in the world, being both concatenative (adding prefixes and suffixes) and non-concatenative (using root-and-pattern derivation) (Al-Sughaiyer & Al-Kharashi, 2004). The landmark work by **Khoja and Garside** developed a root-based stemmer for Arabic that removes prefixes, infixes, and suffixes before using pattern matching against a root lexicon; this stemmer uses several linguistic data files including punctuation, diacritic characters, and a list of 168 stop words (Ali et al., 2019). A light stemmer for classical Arabic by **Thabet** was applied to the Quran and achieved a stemming accuracy of **99.6% for prefix stemming** and **97% for postfix stemming** (Ali et al., 2019). Comprehensive surveys of Arabic stemming consistently identify over-stemming and under-stemming as the two main error classes (Al-Sughaiyer & Al-Kharashi, 2004).

Urdu Stemming

Several stemming systems have been developed for Urdu. The earliest prominent system, **Assas-Band** (Akram et al., 2009), used prefix and suffix removal with exception lists but was reported to be "highly dependent on very large rules lists as well as exception lists," significantly affecting its efficiency (Ali et al., 2019, p. 139). The **Light Weight Stemmer for Urdu** similarly showed low accuracy on prefix removal, achieving only approximately **14.34% average accuracy on prefix rules** in comparative evaluation (Ali et al., 2019).

The most comprehensive system, developed by **Ali, Khalid, and Saleemi (2019)** and published in the *International Arab Journal of Information Technology*, introduced six novel Urdu infix word classes: Alif Arabic Masdar, Te Arabic Masdar, Isam Fiale, Isam Mafool, Arabic Jamah, and Isam Zarf Makaan (Ali et al., 2019). Testing on the Corpus C4 (43,988 total words, 32,388 unique words), the system achieved:

- **Prefix rule accuracy**: 85.60%
- **Infix rule accuracy**: 90.36% (across all infix classes)
- **Postfix rule accuracy**: 89.60%
- **Add-character list accuracy**: 85.26%
- **Overall accuracy**: **90.83%** (Ali et al., 2019)

Importantly, the authors note that "the proposed stemming rules are generic and have the ability to generate the stem of Urdu words as well as loan words belonging to other languages i.e., Arabic, Persian, Turkish" (Ali et al., 2019, p. 145), making this system a natural candidate for integration into a cross-lingual framework.

Persian Stemming

The first Persian stemmer, **Bon** (Tashakori), uses an iterative longest-matching algorithm to remove prefixes and suffixes until a valid stem is produced; it improved recall by 40% (Ali et al., 2019). **Mokhtaripour's** stemmer generates stems without a language dictionary, improving query performance by 46% (Ali et al., 2019). The multi-phase algorithm by **Estahbanati and Javidan (2009)** uses BNF machine rules across 40 steps of suffix and prefix removal, augmented by morphological rule tables, and was found to be more systematic in handling Persian verb morphology. The **bottom-up Persian stemmer** by Khosravi (2008), published at IJCNLP, proposes finding the stem first rather than stripping affixes, and operates without requiring pre-built morphological knowledge (Khosravi, 2008). The open-source **Perstem** tool (Safari, GitHub) combines Persian stemming with morphological analysis, transliteration, and partial POS tagging for Perso-Arabic input encoded in UTF-8 (Safari, 2012). Dolamic and Savoy (2009) further demonstrated that even simple stemming significantly improves IR precision for Persian.

Sindhi Stemming

The thesis by **Mohsin Raza Shah (2016)** at Shah Abdul Latif University the primary foundation of this paper presents the first dedicated Sindhi stemmer using a rule-based stripping approach applied to secondary words. The system achieved an **overall accuracy of 84.85%** (cumulative SER of 15.15%), built on a lexicon of 5,327 words and 38 linguistic rules, with individual SER values of 25.68% for prefix words, 10.16% for suffix words, and 9.61% for prefix-suffix words (Shah, 2016). A **finite-state morphological analyser** for Sindhi (Motlani, Tyers & Sharma, 2016), presented at LREC 2016, achieved 81% naïve corpus coverage and over 97% precision on known tokens using the Apertium platform. Research on **Sindhi morphology** by Sodhar et al. (2023) confirmed that Sindhi morphological complexity stems from "different prefix, suffix, and stem placements in

words," vowel deletion ambiguity, and the non-concatenative structure shared with Arabic-script languages (Sodhar et al., 2023, p. 2899). A large Sindhi neural embedding corpus was documented by Narejo and Mahar (2019) using GloVe, Skip-gram, and CBOW models over 61 million words, providing a strong basis for statistical work.

Cross-Lingual and Unified Approaches

Unified processing of Perso-Arabic scripts for Automatic Speech Recognition was recently explored by researchers demonstrating that a common preprocessing pipeline for Persian, Arabic, and Urdu is feasible and can be extended to Sindhi and Pashto (Towards Unified Processing of Perso-Arabic Scripts, ABJAD-NLP 2025). Cross-lingual transfer learning for low-resource Perso-Arabic languages has also been validated for ASR tasks (Efficient ASR for Low-Resource Languages, IJCNLP 2025 Findings). The Kashmir Journal study on comparative morphology of loan nouns across Urdu, Sindhi, Punjabi, Marwari, and Pashto confirms that "Urdu, Sindhi, and Punjabi share similar morphological patterns in their nominal inflections" (Kashmir Journal of Language Research, 2020). However, no prior work has proposed a unified **stemming** framework that specifically targets Sindhi, Urdu, and Persian simultaneously, establishing a clear research gap that this paper fills.

Comparative Morphological Analysis

Script-Level Commonalities

All three languages use the Perso-Arabic script encoded in Unicode. Sindhi's alphabet comprises **52 letters**, Urdu's **39 letters**, and Persian's **32 letters**, all sharing the same core 28 Arabic letters (Sodhar et al., 2023; Ali et al., 2019; Estahbanati & Javidan, 2009). All three languages are written right-to-left and rarely include diacritical marks in ordinary text, creating systematic morphological ambiguity (Shah, 2016; Motlani et al., 2016). The shared Unicode block (U+0600–U+06FF) makes cross-lingual character processing possible, but character homophones multiple Unicode code points for the same phoneme remain a significant preprocessing challenge (Motlani et al., 2016).

Feature	Sindhi	Urdu	Persian
Alphabet size	52 letters	39 letters	32 letters
Shared Arabic base letters	~28	~28	~28
Unique letters	18	~11	~4
Script direction	Right-to-left	Right-to-left	Right-to-left
Diacritics in ordinary text	Rare	Rare	Rare
Unicode block	U+0600–U+06FF	U+0600–U+06FF	U+0600–U+06FF

Table 1: Script-level comparison of Sindhi, Urdu, and Persian (Shah, 2016; Motlani et al., 2016; Ali et al., 2019)

Morpheme Classes

Shah (2016) identifies that Sindhi secondary words are formed with prefix, stem, and suffix components, while compound words are formed by combining two or more primary words. Ali et al. (2019) document 60 generic Urdu prefixes including نا (na-), بد (bad-), غیر (ghair-), لا (la-), and بے (be-), and a list of 140 suffixes covering gender, plural, and tense marking. Estahbanati and Javidan (2009) similarly document Persian prefix forms including نا (nā-) and می (mi-), and a clear agglutinative suffix system for Persian verbs. The negation prefix نا appears functionally

identically in all three languages, a finding confirmed by the comparative nominal morphology study in the Kashmir Journal of Language Research (2020).

Infix morphology is the most distinctive feature of Urdu, inherited from Arabic root-and-pattern derivation. Ali et al. (2019) introduce six Urdu infix word classes and demonstrate that correctly handling these classes raises overall accuracy from approximately 70% to 90.83%. While Persian and Sindhi have limited native infix morphology, Arabic loan words present in all three languages may carry such infixes, making an infix-handling module essential in any unified framework (Ali et al., 2019; Shah, 2016).

Shared Morpheme Inventory

Through cross-linguistic analysis of the three languages' morphological descriptions, we identified **127 shared morphemes** across Sindhi, Urdu, and Persian:

Morpheme Type	Sindhi	Urdu	Persian	Shared
Prefix morphemes	22	60	41	31
Suffix morphemes	43	140	55	67
Infix patterns	4	38	6	8
Stop-word overlap	~30	~200	~150	21

Table 2: Shared morpheme inventory across three languages (compiled from Shah, 2016; Ali et al., 2019; Estahbanati & Javidan, 2009)

Language-Specific Differences

Despite shared features, critical differences require dedicated language modules:

- **Sindhi** has 18 unique implosive and retroflex consonants (e.g., ٻ, پ, ڏ, ڊ, ڍ) absent in Urdu and Persian, requiring a dedicated orthographic normalization layer (Motlani et al., 2016; Shah, 2016)
- **Urdu** has extensive infix morphology from Arabic, absent or marginal in Sindhi and Persian, requiring the six infix word classes introduced by Ali et al. (2019)
- **Persian** has a transparent agglutinative verb system with well-defined stem-ending distinctions not found in Urdu or Sindhi (Estahbanati & Javidan, 2009)
- Sindhi **compound words** follow different formation rules from Persian compound nouns and Urdu light-verb constructions (Shah, 2016)

The UPASS Framework

Design Principles

UPASS is designed around four core principles drawn from best practices in the existing literature:

1. **Modularity**: A shared core processing pipeline with swappable language-specific extension modules, following the framework approach advocated by Porter (1980) and the Snowball framework
2. **Lexicon-independence in the core**: The core algorithm applies rules first and consults lexica only for exception handling, as validated by Mokhtaripour's dictionary-free Persian stemmer (Ali et al., 2019)
3. **Extensibility**: New Perso-Arabic languages (Pashto, Balochi, Punjabi-Shahmukhi) can be added by contributing new extension modules
4. **Unicode compatibility**: All components use canonical Perso-Arabic Unicode encoding, consistent with the Apertium-based Sindhi finite-state analyser (Motlani et al., 2016)

System Architecture

UPASS comprises six sequential components:

text

Input Text
  |
  ▼
[1] Unicode Normalization & Script Detection
  |
  ▼
[2] Tokenization & Stop-Word Removal
  |
  ▼
[3] Minimum Word Length Filter (≤3 characters → return as stem)
  |
  ▼
[4] Cross-Lingual Morpheme Matching Engine
  | ├── Shared Prefix Module (31 shared prefixes + lang-specific)
  | ├── Infix Module (Urdu: 6 classes; Sindhi/Persian: Arabic loan infixes)
  | └── Shared Suffix Module (67 shared suffixes + lang-specific)
  |
  ▼
[5] Language-Specific Extension Module
  | ├── Sindhi: Implosive normalization + compound decomposition
  | ├── Urdu: Full 6-class infix rules + Add-Character Lists
  | └── Persian: Agglutinative verb stem identification
  |
  ▼
[6] Stem Validation via Language Lexicon + Add-Character Normalization
  |
  ▼
Output: Stem

The minimum word length rule in step 3 is adapted from Ali et al. (2019), who demonstrate that "an Urdu word comprising only two or three characters is already a stem word" (p. 139); this rule was validated as equally applicable in Sindhi by Shah (2016) and in Persian by Estahbanati and Javidan (2009).

Cross-Lingual Morpheme Repository (CLMR)

The CLMR is the shared resource at the heart of UPASS. It contains:

- **127 shared morphemes** with language applicability flags (S = Sindhi, U = Urdu, P = Persian)
- **Morpheme priority weights** based on frequency within each language's corpus
- **Exception lists** compiled from: the Urdu PrGEL of ~5,000 words and InGEL of ~3,000 words (Ali et al., 2019), the Sindhi 5,327-word lexicon (Shah, 2016), and Persian exception entries derived from Estahbanati and Javidan (2009)
- **Add-character lists** for stem normalization: eight character-specific lists for Urdu developed by Ali et al. (2019) extended to Sindhi and Persian forms

Unicode Normalization Module

Motlani et al. (2016) document that the same Sindhi phoneme can appear in multiple Unicode code points depending on a letter's position in a word. UPASS resolves this through a **156-entry normalization table** mapping all Perso-Arabic presentation forms to canonical Unicode code points before any morphological processing. This step is critical for achieving consistent matching against the CLMR and language lexica.

Algorithm Design

UPASS Core Algorithm

The UPASS algorithm for each input token *w* proceeds as follows:

Step 1 — Unicode Normalization: Normalize all characters in *w* using the 156-entry normalization table (Motlani et al., 2016).

Step 2 — Stop-Word Filtering: Compare *w* against the cross-lingual stop-word list. Ali et al. (2019) use a static list of 200 Urdu stop words; Shah (2016) identifies similar stop words for Sindhi. The combined cross-lingual list retains 21 language-shared stop words. If matched, discard *w*.

Step 3 — Minimum Word Length Rule: If $|w| \leq 3$, mark *w* as its own stem and return. This rule is validated by Ali et al. (2019) for Urdu and consistent with Shah's (2016) implementation for Sindhi.

Step 4 — Prefix Removal: Search *w* against CLMR prefix module using longest-match first. If a prefix is found and the residual stem length ≥ 3, remove it. Apply language-specific prefix rules; for Urdu, the 60 generic prefix rules of Ali et al. (2019) are used. For Sindhi, the 38 linguistic rules of Shah (2016) guide prefix removal.

Step 5 — Infix Detection: If language is Urdu, apply the six infix word class rules: Alif Arabic Masdar, Te Arabic Masdar, Isam Fiale, Isam Mafool, Arabic Jamah, and Isam Zarf Makaan (Ali et al., 2019). For Sindhi and Persian, check only for Arabic loan-word infixes catalogued in the CLMR infix module.

Step 6 — Suffix Removal: Apply CLMR suffix module using longest-match. For Urdu, the 140-suffix list of Ali et al. (2019) is applied. For Sindhi, gender and case suffix rules from the 38-rule repository of Shah (2016) are applied. For Persian, Estahbanati and Javidan's (2009) BNF-based suffix chain rules apply.

Step 7 — Add-Character Normalization: After stripping, check the resulting stem against the add-character lists to produce a valid stem form. Ali et al. (2019) developed eight character-specific add-character lists (for he, ya, wao, alif, and combinations) to handle cases where stripping leaves an incomplete stem.

Step 8 — Stem Validation: Check the resulting stem against the language-specific lexicon. Sindhi uses the 5,327-word lexicon (Shah, 2016); Urdu uses the 10,000-word stem dictionary (Ali et al., 2019); Persian uses the Bon stem dictionary (Tashakori, as cited in Ali et al., 2019). If not found, apply backtracking.

Step 9 — Return Stem.

Pseudocode

text

```
FUNCTION UPASS_Stem (w, lang):
  w ← Normalize_Unicode (w, normalization_table_156)// Motlani et al., 2016
  IF IsStopWord (w, CLMR.stop_words[lang]) THEN RETURN null
```

```
IF Length(w) ≤ 3 THEN RETURN w// Ali et al., 2019
w ← CheckPrGEL (w, CLMR.prefix_exceptions[lang])
IF NOT in_PrGEL:
   w ← RemoveLongestPrefix (w, CLMR.prefixes[lang])// Shah, 2016; Ali et al., 2019
IF lang = "Urdu":
   w ← RemoveInfix_SixClasses (w, InfixClasses)// Ali et al., 2019
ELSE:
   w ← RemoveArabicLoanInfix (w, CLMR.infixes)
w ← RemoveLongestSuffix (w, CLMR.suffixes[lang])
IF lang = "Sindhi":
   w ← ApplySindhiSecondaryRules(w)// Shah, 2016
IF lang = "Persian":
   w ← ApplyPersianVerbStemRules(w)// Estahbanati & Javidan, 2009
w ← ApplyAddCharList (w, CLMR.add_chars[lang])// Ali et al., 2019
IF ValidStem (w, Lexicon[lang]) THEN
   RETURN w
ELSE
   RETURN BacktrackBestStem(w)
```

Evaluation Methodology

Corpora

Three corpora are used for evaluation:

Language	Corpus	Total Tokens	Unique Tokens	Source
Sindhi	Sindhi Secondary Word Corpus	86,733	50,327	Shah (2016)
Urdu	Headline News Corpus C4	43,988	32,388	Ali et al. (2019)
Persian	Multi-genre Persian Text	~50,000	~28,000	Estahbanati & Javidan (2009)

Table 3: Evaluation corpora

Evaluation Metrics

Following established practice in stemming evaluation (Shah, 2016; Ali et al., 2019), we use:

- **Stemmed Error Rate (SER)**: $SER = \frac{\text{Incorrectly Stemmed Words}}{\text{Total Tested Words}} \times 100$
- **Accuracy**: $Accuracy = 100 - SER$
- **Precision**: Correct stems produced / Total stems produced
- **Recall**: Correct stems produced / Total correct stems in gold standard

Baseline Systems

UPASS is compared against:

- Sindhi Rule-Based Stemmer, Shah (2016), with cumulative SER of 15.15% and accuracy of 84.85%
- Comprehensive Urdu Stemmer, Ali et al. (2019), with overall accuracy of 90.83%
- Multi-Phase Farsi Stemmer, Estahbanati and Javidan (2009), with cumulative SER of ~13.6%

Results and Discussion

Sindhi Stemming Results

Word Class	Baseline SER (Shah, 2016)	UPASS SER	Accuracy Gain
Prefix words	25.68%	22.14%	+3.54%
Suffix words	10.16%	8.91%	+1.25%
Prefix-Suffix words	9.61%	7.88%	+1.73%
Cumulative	**15.15%**	**12.97%**	**+2.18%**

Table 4: Sindhi stemming comparison (baseline from Shah, 2016)

UPASS improves on the baseline Sindhi stemmer (Shah, 2016) primarily through the shared prefix module, which captures 11 additional Perso-Arabic-origin prefixes present in Sindhi text due to extensive borrowing from Urdu and Persian. Shah (2016) documented 38 linguistic rules for Sindhi stemming; the CLMR supplements these with 31 cross-lingual shared prefixes validated by Ali et al. (2019) and Estahbanati and Javidan (2009).

Urdu Stemming Results

Rule Type	Baseline Accuracy (Ali et al., 2019)	UPASS Accuracy
Prefix rules	85.60%	86.20%
Infix rules (all classes)	90.36%	90.36% (unchanged)
Postfix rules	89.60%	90.44%
Add-Character Lists	85.26%	86.01%
Overall	**90.83%**	**91.42%**

Table 5: Urdu stemming comparison (baseline from Ali et al., 2019)

UPASS maintains near-parity with the state-of-the-art Urdu stemmer (Ali et al., 2019) while adding the cross-lingual normalization layer. The infix module is preserved entirely from Ali et al. (2019) as it represents the current state of the art. The marginal gains in prefix and postfix accuracy derive from the unified Unicode normalization step, which resolves character homophone ambiguities particularly prevalent in cross-script Urdu documents.

Persian Stemming Results

Phase	Baseline SER (Estahbanati & Javidan, 2009)	UPASS SER
Noun stemming	14.2%	13.1%
Verb stemming	12.8%	11.4%
Cumulative	**13.6%**	**12.3%**

Table 6: Persian stemming comparison (baseline from Estahbanati & Javidan, 2009)

Persian benefits most from the add-character normalization lists developed by Ali et al. (2019) for Urdu, which equally apply to Persian-origin loan words present across all three corpora. The agglutinative Persian verb stem module retains the BNF-machine rules from Estahbanati and Javidan (2009).

Unified Cross-Language Performance

Language	Baseline SER	UPASS SER	UPASS Accuracy
Sindhi	15.15% (Shah, 2016)	12.97%	87.03%
Urdu	9.17% (Ali et al., 2019)	8.58%	91.42%
Persian	13.60% (Estahbanati & Javidan, 2009)	12.30%	87.70%
Average	**12.64%**	**11.28%**	**88.72%**

Table 7: Unified cross-language performance of UPASS

Discussion

The key insight is that **prefix sharing yields the highest cross-lingual gains**, particularly for Sindhi. Shah (2016) noted that "Sindhi language is well versed regarding its persistence and stability specifically while borrowing and merging of words from other languages" this borrowing directly introduces Urdu and Persian prefix patterns into Sindhi vocabulary that the original 38-rule Sindhi repository did not fully cover (Shah, 2016, p. 9).

The **infix module** validated by Ali et al. (2019) remains Urdu-specific in its full six-class form, but its Arabic loan-word subset applies to Sindhi and Persian text. Ali et al. (2019) explicitly state their rules "have the ability to generate the stem of Urdu words as well as loan words belonging to other languages i.e., Arabic, Persian, Turkish" (p. 145), confirming cross-lingual utility.

The main residual error sources across all three languages are:

1. **Diacritics ambiguity**: All three languages rarely include diacritical marks, leading to multiple valid morphological analyses (Motlani et al., 2016; Shah, 2016)
2. **Homographs**: Same orthography, different roots (e.g., homographic words in Sindhi documented by Shah, 2016, and Motlani et al., 2016)
3. **Compound word decomposition**: Least accurate sub-task in Shah's (2016) Sindhi stemmer, and similarly under-researched for Persian and Urdu compounds
4. **Verb paradigm coverage**: The Sindhi verb paradigm is "not fully enumerated in existing resources" (Motlani et al., 2016, p. 3)

Conclusion and Future Work

This paper has proposed **UPASS**, the first unified stemming framework for Sindhi, Urdu, and Persian three major languages sharing the Perso-Arabic script. Built on a systematic comparative morphological analysis identifying 127 shared morphemes, a modular six-component algorithmic pipeline, and a cross-lingual morpheme repository, UPASS achieves an average accuracy of **88.72%** across the three languages, consistently outperforming language-specific baselines by an average of 1.36% while enabling resource sharing.

Theoretical contributions include:

- The first formal comparative morphological analysis of Sindhi (Shah, 2016), Urdu (Ali et al., 2019), and Persian (Estahbanati & Javidan, 2009) focused on stemming
- A documented inventory of 127 shared prefix, suffix, and infix morphemes
- Validated cross-lingual exception lists and add-character normalization tables

Future directions include:

- **Deep learning extension**: Replacing rule-based stripping with a neural sequence-to-sequence model trained on UPASS-labeled data (Sodhar et al., 2023 recommend this as a direction for Sindhi NLP)
- **Extension to other Perso-Arabic languages**: Pashto, Balochi, Punjabi-Shahmukhi, and Kashmiri all use variants of the Perso-Arabic script and share morphological features with the three languages studied here
- **Integration with full NLP pipeline**: As Shah (2016, p. 7) notes, the stemmer is "the backbone process of any IR system" integrating UPASS into a complete Sindhi, Urdu, and Persian IR system is the logical next step

References

Akram, Q., Naseer, A., & Hussain, S. (2009). Assas-Band: An affix-exception-list based Urdu stemmer. *Proceedings of the 7th Workshop on Asian Language Resources (ALR7)*, Association for Computational Linguistics, Suntec, Singapore, pp. 40–47.

Al-Sughaiyer, I. A., & Al-Kharashi, I. A. (2004). Arabic morphological analysis techniques: A comprehensive survey. *Journal of the American Society for Information Science and Technology*, 55(3), 189–213.

Ali, M., Khalid, S., & Saleemi, M. (2014). A novel stemming approach for Urdu language. *Journal of Applied Environmental and Biological Sciences*, 4(7S), 436–443.

Ali, M., Khalid, S., & Saleemi, M. (2019). Comprehensive stemmer for morphologically rich Urdu language. *The International Arab Journal of Information Technology*, 16(1), 138–147. Retrieved from https://www.iajit.org/portal/PDF/January%202019,%20No.%201/11296.pdf

Bacchin, M., Ferro, N., & Melucci, M. (2002). Experiments to evaluate a statistical stemming algorithm. *Proceedings of the CLEF 2002 Workshop on Monolingual Information Retrieval*, Rome, pp. 161–168.

Bento, C., Cardoso, A., & Dias, G. (2005). Progress in artificial intelligence. *Proceedings of the 12th Portuguese Conference on Artificial Intelligence (EPIA 2005)*, Covilhã, Portugal, pp. 693–701.

Dawson, J. L. (1974). Suffix removal and word conflation. *ALLC Bulletin*, 2(3), 33–46.

Dolamic, L., & Savoy, J. (2009). Persian language, is stemming efficient? *Proceedings of the 9th International Workshop on Timely Information Retrieval (TIR-09)*, pp. 1–8. Retrieved from https://webis.de/events/tir-09/tir09-papers-final/dolamic09-persian-language-is-stemming-efficient.pdf

Estahbanati, S. H. G., & Javidan, R. (2009). A new multi-phase algorithm for stemming in Farsi language. *International Journal of Computer Theory and Engineering (IJCTE)*, 1(3). Retrieved from https://www.ijcte.org/papers/381-JG524.pdf

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., & Tyers, F. M. (2011). Apertium: A free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2), 127–144.

Husain, S. (2004). A rule-based Sindhi stemmer. *Unpublished manuscript*. Center for Language Engineering, Lahore, Pakistan.

Kashmir Journal of Language Research. (2020). Comparative morphological analysis of loan nouns in Urdu, Sindhi, Punjabi, Marwari, and Pashto. *Kashmir Journal of Language Research*, 23(2). Retrieved from https://kjlr.pk/index.php/kjlr/article/download/84/42/120

Khoja, S., & Garside, R. (1999). Stemming Arabic text. *Lancaster University, Computer Science Department*. Technical Report.

Khosravi, H. (2008). A bottom-up approach to Persian stemming. *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP 2008)*, Hyderabad, India, pp. 583–588. Retrieved from https://aclanthology.org/I08-2076.pdf

Krovetz, R. (1993). Viewing morphology as an inference process. *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Pittsburgh, PA, pp. 191–202.

Lovins, J. B. (1968). Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11(1–2), 22–31.

Mahar, J. A., & Memon, G. Q. (2010). Rule based part of speech tagging of Sindhi language. *International Conference on Signal Acquisition and Processing*, 134, 101–106.

Mokhtaripour, A. (2006). A new Persian stemmer. *Unpublished manuscript*, Sharif University of Technology, Tehran, Iran. Cited in Ali et al. (2019).

Motlani, R., Tyers, F. M., & Sharma, D. M. (2016). A finite-state morphological analyser for Sindhi. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Portorož, Slovenia, pp. 2572–2577. European Language Resources Association (ELRA). Retrieved from http://www.lrec-conf.org/proceedings/lrec2016/pdf/1124_Paper.pdf

Narejo, N., & Mahar, J. A. (2019). Morphology: Sindhi morphological analysis for natural language processing. *Mehran University Research Journal of Engineering and Technology*. Semantic Scholar. Retrieved from https://www.semanticscholar.org/paper/Morphology:-Sindhi-morphological-analysis-for-Narejo-Mahar/fe5f551b0473edd83a7e1836d5bcf6d

Paice, C. D. (1990). Another stemmer. *ACM SIGIR Forum*, 24(3), 56–61.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program: Electronic Library and Information Systems*, 14(3), 130–137.

Rahman, M. U., & Bhatti, M. I. (2010). Finite state morphology and Sindhi noun inflections. *Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation (PACLIC 24)*, Tohoku University, Japan, pp. 669–676.

Rahman, M. U. (2010). Towards Sindhi corpus construction. *Conference on Language and Technology*, Lahore, Pakistan.

Safari, J. (2012). Perstem: Persian stemmer and morphological analyzer [Software]. GitHub. Retrieved from https://github.com/jonsafari/perstem

Shah, M. R. (2016). *Stemmer of Sindhi secondary words using rule-based stripping approach for information retrieval system* (Master's thesis). Department of Computer Science, Shah Abdul Latif University, Khairpur Mirs, Sindh, Pakistan.

Sodhar, I. N., Sulaiman, A., & Buller, A. H. (2023). Morphology-assisted Sindhi text analysis for natural language processing. *Indian Journal of Science and Technology*, 16(35), 2898–2906. https://doi.org/10.17485/IJST/v16i35.1719. Retrieved from https://sciresol.s3.us-east-2.amazonaws.com/IJST/Articles/2023/Issue-35/IJST-2023-1719.pdf

Tashakori, M. (2002). *Bon: A Persian stemmer* (Unpublished). Sharif University of Technology, Tehran, Iran. Cited in Ali et al. (2019).

Thabet, N. (2004). Stemming the Qur'an. *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, COLING 2004, Geneva, Switzerland, pp. 28–31.

Towards Unified Processing of Perso-Arabic Scripts for ASR. (2025). *Proceedings of the ABJAD-NLP Workshop, IJCNLP-AACL 2025*. Association for Computational Linguistics. Retrieved from https://aclanthology.org/2025.abjadnlp-1.3.pdf

Efficient ASR for Low-Resource Languages: Leveraging Cross-Lingual Transfer Learning. (2025). *Findings of IJCNLP-AACL 2025*. Association for Computational Linguistics. Retrieved from https://aclanthology.org/2025.findings-ijcnlp.99.pdf

Xuanli, H., Ali, M., Khalid, S., & Saleemi, M. (2024). An extended pattern based comprehensive stemmer for the Urdu language. *ACM Transactions on Asian and Low-Resource Language Information Processing*.
https://doi.org/10.1145/3701231. Retrieved from https://dl.acm.org/doi/pdf/10.1145/3701231