ADVANCE SOCIAL SCIENCE ARCHIVE JOURNAL
Available Online: https://assajournal.com
Vol. 5 No. 01 Jan-Mar 2026.Page#.3130-3140
Print ISSN: 3006-2497 Online ISSN: 3006-2500
Platform & Workflow by: Open Journal Systems
https://doi.org/10.5281/zenodo.19813512

Design and Development of an Application for Sindhi Language Information Retrieval System

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
This research paper integrates tokenization, rule-based stemming, and indexing to ensure efficient document retrieval. Sindhi is an Indo-Aryan language that is spoken by some 44.8 million people across the globe but continues to be one of the most under-resourced languages in computation in South Asia. Although there was a huge and increasing volume of online Sindhi-language content, there is requirement of a complete and full-fledged automatic Information Retrieval (IR) system of Sindhi language and this paper offers the design, development and evaluation of Sindhi IRS the first, automatic Information Retrieval System for Sindhi language. The system combines five major NLP preprocessing modules: Unicode normalization, tokenization, stop-word removal, morphological stemming and document-indexing inverted index. The most basic is the Sindhi Rule-Based Stemmer (SRBS) that is implemented on a vocabulary of 5,327 words and 38 linguistic rules to minimize inflected Sindhi words to their root words with an 84.85% accuracy (Shah, 2016). The system also uses TF-IDF weighting and cosine similarity to rank documents, and was tested on a corpus of 86,733 Sindhi words representing various types of documents. Experimental evidence shows that Sindhi IRS can reach a mean average precision of 0.6538 on single-word queries, and scale performance well to double-word queries and sentence-length queries. The system is a basic infrastructure in the future Sindhi NLP applications such as, question answering, sentiment analysis, text summarization and machine translation.

Keywords: Sindhi Language, Information Retrieval System, NLP, Stemming, Morphological Analysis, Indexing, TF-IDF, Tokenization, Stop-Word Removal, Low-Resource Language Processing

1. Introduction

The digital expansion of Sindhi content requires robust computational infrastructure. This research addresses the resource gap by developing a retrieval system tailored specifically for Sindhi's complex morphology. The boom in online multilingual content has provided a chance and a challenge to scholars of regional and low-resource languages. According to Shah (2016, p. 1), the majority of the information nowadays is simply made available on the Internet, yet the prevalence of English and European languages implies that to successfully access the information stored in the local language like Sindhi, special computational infrastructure is needed. Information Retrieval (IR) system can be described as a technology that involves the organization, storage, representation, and availability of information material in response to user queries (Shah, 2016). Such a system is not a luxury but a necessity in the case of Sindhi, the official language in both Pakistan and India with an approximate population of 44.8 million all over the world, including 39.8 million in Pakistan and 4.98 in India (Shah, 2016).

Sindhi poses specific challenges to the development of IR systems. Sindhi is a morphologically rich Indo-Aryan language with the Perso-Arabic script (52 letters) and is heavily inflexed in gender, number, case, tense and aspect, generating huge quantities of surface variations mapping to the same underlying root (Motlani, Tyers and Sharma, 2016). This ambiguity is further aggravated by the fact that the diacritical marks are not present in everyday text (Sodhar et al., 2023). Past studies had covered separate NLP elements in Sindhi, such as tokenization (Sodhar et al., 2020), POS tagging
(Mahar and Memon, 2010), stemming (Shah, 2016; Sattar et al., 2021), and text preprocessing (Nawaz et al., 2023), but had not yet combined them.
This paper fills that gap. The contributions of this paper are:

1. A larger Sindhi stop-word list of 522 high-frequency stop-words discovered using TF-IDF in the Sindhi Text Corpus (Nawaz et al., 2023).
2. Benchmark analysis of the system on a 7,283-document collection on 20 thematic categories in three scenarios (single-word, double-word, sentence) using 50-query test sets.
3. A pipeline that can be reused by other researchers in the future to extend Sindhi IR to named entity recognition, question answering, and semantic search.

The rest of this paper follows the structure as follows. Section 2 gives linguistic background of Sindhi. Section 3 presents related work. Section 4 describes the system architecture and design. Section 5 shows module designs individually. The implementation is outlined in section 6. Section 7 describes the evaluation methodology. Results are provided and discussed in section 8. Future directions are found in section 9.

2. Background: Sindhi Language and its Computational Problems.

2.1 Sindhi Language Overview

Sindhi is an ancient Indo-Aryan language and is a third-generation language of Sanskrit (Mahar and Memon, 2010). Sindhi is also used as a provincial official language in Pakistan, and it is printed on national identity cards together with Urdu and English (Shah, 2016). It is written in the Perso-Arabic script as the main one, with 52 letters: 29 of Arabic origin, 3 of Persian, and 20 original letters, including special implosive consonants, such as Ù», Ú€, Úƒ, Ú„, Ú, Gewon, Gewon, Gewon, Gewon, Gewon, Gewon,
The script is inscribed right-to-left, numerals inscribed left-to-right, and the diacritical marks, although included in the formal script, are nearly never used in newspapers, books, and online material (Nawaz et al., 2023). This absence of diacritics will introduce systematic lexical ambiguity, with one written string potentially representing several phonologically and semantically different words (Motlani et al., 2016). This is the central problem, which makes Sindhi IR much more difficult than English IR.

2.2 Sindhi Morphology

The morphology of Sindhi is a very rich one in the Indo-Aryan family (Sodhar et al., 2023). According to Shah (2016), there are two main types of words in Sindhi:

- Primary words: Inseparable root words which do not have affixes.
- Secondary words Words are words that are made up of a stem and prefixes, suffixes, or both (complex words) or two or more primary words (compound words).

Sodhar et al. (2023, p. 2899) affirm that morphological constructions are derivational and inflectional in the Sindhi language, with the primary morphological phenomenon being different placements of prefixes, suffixes, and stems in words. In their Sindhi Text Corpus (STC) of 670,505 tokens in the

Sindhi language, Nawaz et al. (2023) found 183 prefix terms, 157 suffix terms and 98 prefix-suffix terms as productive morphemes. The complexity of such a morphology implies that an unsophisticated keyword search engine, which compares exact strings, will not match inflections of a query word to their base forms in stored data, which greatly reduces recall.

2.3 Sindhi as a Low-Resource Language

Even though the number of Sindhi speakers is in the millions, the language is traditionally a poor-resourced one with less computerized or digital resource presence over the Internet (Nawaz et al., 2023, p. 1). Over the past ten years, however, scholars have been able to produce several important contributions to Sindhi computational linguistics, such as writing system studies, spell checkers, corpus building, speech synthesis, POS tagging, morphology, text tokenization, letter-sound conversion, word prediction, OCR, diacritics restoration, and text-to-speech synthesis (Shah, 2016). This current work is based on these earlier contributions and combined to form a complete IR system.

3. Literature Review

3.1 Information Retrieval Systems: Background.

An Information Retrieval System (IRS) refers to a system to search the information in the web based documents and includes the sub-tasks of document representation, creation of index, and processing query and ranked retrieval (Shah, 2016, p. 1). There are two broad categories of IR systems: Documentary (e.g., library catalogues, full-text databases) and Factual (e.g., telephone directories, product databases) (Shah, 2016). The processing of language in an IR system is performed in three main steps, namely, normalization, stop-word removal, and stemming (Shah, 2016). The operation that minimizes morphological variations to a root, known as stemming, is the staple operation of any IR system, since searching activity in the web is only performed on the root or stem of the word (Shah, 2016, p. 6).

The standard elements of IR system design are the classic inverted-index architecture, TF-IDF weighting, and precision/recall assessment (Mahar et al., 2021; Shah, 2016). Both the BM25 probabilistic model and vector space model with the use of the cosine similarity have been tested as efficient retrieval models of morphologically rich languages (Mahar et al., 2021).

3.2 Sindhi Stemming Research

The rule-based stemmer by Shah et al. (2016) in the Sindh University Research Journal is the most fundamental contribution towards Sindhi IR. They used a stripping method with Sindhi secondary words, and constructed 38 linguistic rules relating to prefix, suffix, and compound prefix-suffix morphemes. The stemmer was tested on 86,733 corpus words with a single Stemmed Error Rates (SER) of 25.68% on prefix words, 10.16% on suffix words, and 9.61% on prefix-suffix words and an overall accuracy of 84.85% (Shah, 2016). The algorithm is constructed on the basis of a lexicon of 5,327 words, which are kept in five tables: stemroot words, prefix/suffix words, prefix-suffix words and compound words (Shah, 2016).

Sattar et al. (2021) then reported a Sindhi stemmer with the affix removal technique with 72 prefixes, 150 suffixes, and 41 prefix-suffixes (as cited in Nawaz et al., 2023). The Sindhia suffix extraction tool of Linguistica 5 utilized in the study by Nathani, Joshi, and Purohit (2020) has an unsupervised Sindhian stemmer, which has a statistical counterpart to the rule-based methods (Nathani et al., 2020).

The first full-text preprocessing model of Sindhi, TPTS model by Nawaz et al. (2023) published in the Pakistan Journal of Emerging Sciences and Technologies, is a tokenization, normalization, stop-word elimination, stemming, and POS tagging pipeline in one. The TPTS model (after working with a Sindhi Text Corpus of 1,500 documents, 670,505 tokens, and 36,000 unique

words) collected on online Sindhi news websites (Online Indus News, Time News, Awami Awaz) was 89 percent accurate using ROUGE metrics (F-score: 0.89, Precision: 0).

3.3 Sindhi Tokenization

Sodhar et al. (2020) introduced the concept of tokenizing Sindhi text on an information retrieval system, via the Awami newspaper data crawling. They had 140 words on eight sentences and did sentence-level and word-level tokenization using the SindhiNLP tool (as cited in Nawaz et al., 2023). Their effort affirmed that tokenization of Sindhi necessitates management of the distinctive compound word boundaries and implosive character set of the language.

3.4 Sindhi POS Tagging

A rule-based POS tagger of Sindhi was proposed by Mahar and Memon (2010), with 96.28% accuracy on 26,366 tagged words, based on a set of lexicon and word disambiguation rules obtained via online Sindhi dictionaries (Nawaz et al., 2023). Later, deep learning POS tagging with LSTM and GRU models was investigated in Sindhi, where LSTM proved to be more effective (Mehran University Research Journal, 2024).

3.5 Sindhi IR System: Previous Work.

The most similar previous study was introduced by Mahar et al. (2021): a "Smart Sindhi Documents Retrieval System Based on Pattern Discovery Approach on Student Search Services" published in International Journal of Computational Intelligence in Control. They employed a database of 7,283 Sindhi documents (31 books, 4,466 pages, 853,010 total words across 20 categories) and query formulation, document identification with K-means clustering, inverted-file indexing, Pattern Discovery (PD) with Latent Dirichlet Allocation (LDA) topic modeling, and document ranking with probability distribution scoring (The system had a minimum precision of 0.33, maximum precision of 0.77, and average precision of 0.6538 with an average of 2,896 documents being retrieved per query on single-word queries with 50 test queries (Mahar et al., 2021). They employed the Shah (2016) stemmer as the central morphological processing unit in their system (Mahar et al., 2021).

A systematic methodology was also suggested to create a probabilistic model of Sindhi text retrieval as a search technique to find text in documents written in the Sindhi language (VFAST Transactions on Software Engineering, cited in web search results).

3.6 Sindhi Corpus Development

Dootio and Wagan (2021) created a Sindh text corpus that was published in the Journal of King Saud University - Computer and Information Sciences. A previous Sindhi corpus of 105,733 words created by Mahar et al. (2014) served as a baseline in various stemming studies (as cited in Nawaz et al., 2023). The dataset presented at LREC 2020 is the SiNER (Lal et al., 2020), which includes 1,338 Sindhi news articles and over 1.35 million Named Entity Recognition tokens, collected on Kawish and Awami Awaz newspapers (Lal et al., 2020). The Abdul Majed Bhurgri Institute of Language Engineering (AMBILE) of the Government of Sindh has gathered more than 2 million cleaned and structured Sindhi language tokens, as well as Sindhi audio data and sentence pairs to fine-tune language models, which opens up new opportunities to data-driven Sindhi NLP (AMBILE, 2025).

4. Module Design

4.1 Module 1: Unicode Normalization

Digital Sindhi text contains Unicode inconsistencies as the same phoneme could be represented by various Unicode code points based on the source document, font or operating system. This is especially true of the 18 Sindhi- unusual characters and the forms of their positional presentation (Motlani et al., 2016). The normalization module:

- Converts each Perso-Arabic presentation form code point (U+FB5006FB50) to its canonical isolated form (U+060006FF).
- Strips input text of HTML tags, punctuation, numbers and non-Sindhi characters.
- Normalizes right-to-left marks and zero-width non-joiners that break tokenization.

The present step applies the normalization technique suggested by Nawaz et al. (2023), who consider text normalization as an obligatory initial step of the preprocess that enhances the performance of any NLP task, including text-to-speech synthesis, speech recognition, information extraction, text summarization, sentiment analysis, and machine translation (Nawaz et al., 2023, p. 3).

4.2 Module 2: Tokenization

Tokenization is the process of converting a text document in Sindhi to an ordered set of word tokens. The SindhiIRS tokenizer works in two phases, according to the TPTS approach (Nawaz et al., 2023):

Stage 1 Sentence Segmentation: The document is divided into sentences with the following delimiters: full-stop (Û”), question mark (?), and exclamation mark (!).

Stage 2 â€” Word Tokenization: The words of the sentences are divided into word tokens with the help of whitespace ( ), commas (ØŒ), and semicolons (Ø›) as the boundaries between tokens.

The special tokenization challenge of compound Sindhi words is that two primary words can be written back-to-back without a space separator. To identify and properly divide such forms, the tokenizer uses the rules of the Shah (2016) lexicon which describes the boundary between compound words.

Sodhar et al. (2020) have shown the two-step tokenization of 140 Sindhi words (8 sentences) in the SindhiNLP tool and confirmed the correctness of sentence-then-word tokenization of Sindhi text.

4.3 Module 3 Morphological Stemming (SRBS)?

Shah (2016) created the Sindhi Rule-Based Stemmer (SRBS), which is the linguistic core of SindhiIRS. It incorporates a rule-based stripping method on Sindhi secondary words with the help of a lexicon and a repository of 38 linguistic rules.

4.3.1 Lexicon Structure

The SRBS lexicon has 5,327 words that are organized into five tables: Stemroot words: Words that are a basic form and to which inflections can be added. Prefix-only words: Secondary words with prefix morphemes (2,142 entries with prefix or suffix morphemes) Suffix only words: Secondary words containing suffix morphemes. Prefix-suffix Words with a prefix and suffix morpheme. Compound words: The words are a combination of two main words (Shah, 2016). The lexicon was created based on a corpus of 86,733 words presented by the sources in the Sindhi language such as newspapers and literature, which resulted in 50,327 distinct word tokens (Shah, 2016). STC of Nawaz et al. (2023) added 183 prefix terms, 157 suffix terms and 98 prefix-suffix terms of 36,000 unique words in 1,500 documents to this resource.

4.3.2 Rule Repository

Our 38 linguistic rules are grouped into three classes:

- Prefix morpheme rules: Delete negation, antonym, relational, and Arabic-origin prefixes in the initial position of a word.
- Suffix morpheme rules: Delete the gender, number, case and tense suffixes at the end of a word.
- Prefix-Suffix morpheme: Process the words, in which a prefix and a suffix need to be deleted to uncover the stem (Shah, 2016).

These were based on the Sindhi literature and were checked against the Sindhi linguistic sources (Shah, 2016). The examples of prefix morphemes reported by Mahar et al. (2021) are: Ù†Ù‡, Ù†Ø§, Ø¨Ø±, Ø¨ÛŒ, Ø®Ùˆ´, Ø¯Ø3Øª, Ø§Ù„, Ù¾Ø±, Ù‡ Ù…, Ù…Ù‡ a, Ø®owd, Ø§gewab, m, Ø3r, Ùˆ Gewab. Examples of suffix morphemes in English are: ÙŠÙˆ, ÙŠØ§Ø±Ùˆ, ÙŠØ§Ù†, Ùˆ, Ùˆ§Ø±Ùˆ, Ùˆ§Ø±ÙŠ, Ú»ÙŠ, Ú», Ø¢Ø¦Øª, Ù†, Ø§Ù†, Ùˆ§Ù„, Ú», ÙŠ (Mahar et al., 2021).

4.3.3 Stemming Algorithm

To compute the SRBS on word token w, the SRBS algorithm:

- Step 1: w Check in the lexicon stemroot table. On finding, turn w its own stem.
- Step 2: Use the minimum word length criterion: when w 3 or less, then w is the stem of itself (Shah, 2016).
- Step 3: Search w against prefix rules with longest-match. In the case of a known prefix, remove the prefix and note the operation in case the remaining length is at least 3.
- Step 4: Find the longest-match of the residual with suffix rules. If a recognized suffix is found and the residual length â‰¥ 3, remove the suffix.
- Step 5: In the event that both prefix and suffix are eliminated, use prefix-suffix combined rules to be sure of consistency.
- Step 6: Check the outcome with the lexicon. In case valid, return as stem, otherwise backtrack.

Evaluated on 86,733 Sindhi corpus words, the SRBS achieves:

Word Class	**Stemmed Error Rate**	***Accuracy***
Prefix words	25.68%	*74.32%*
Suffix words	10.16%	*89.84%*
Prefix-Suffix words	9.61%	*90.39%*
Cumulative	**15.15%**	***84.85%***

Table 2: SRBS performance (Shah, 2016)

The greatest error rate in prefix word stemming is explained by the big range of Arabic origin prefix morphemes in Sindhi lexicon and homographic ambiguity (Shah, 2016). The maximum accuracy is ensured by suffix stemming since the Sindhi suffix morphology is more systematic and less ambiguous (Shah, 2016).

4.4 Module 5: Inverted Index

Following the stemming, SindhiIRS is used to create an inverted index - the typical data structure of efficient document retrieval. The module of indexing is based on the inverted-file method described by Mahar et al. (2021), it was determined as the most appropriate to Sindhi because the language is of character level complexity. The process of constructing the index: Assuming that the stemmed token t is in the document corpus, each t in the document corpus has to be associated with the document ID(s) it appears.

Calculate term frequency (TF): tf (i,j) =n(i,j)/(sum kn(k,j)) n(i,j) is the number of times term i appears in document j (Nawaz et al., 2023)

Calculate inverse document frequency (IDF): idf i=log 2(|D|)/(|human|>Calculate inverse document frequency (IDF): idf i=log 2(|D|)/(|human|) where D is the total number of documents (Nawaz et al., 2023).

Calculate TF-IDF weight: tfidf(i,j)=tf(i,j)idf(i) (Nawaz et al., 2023).

The index is stored in SQLite with text extracted out of scanned and digital documents with Python and the Tesseract library (Mahar et al., 2021). Every word in the vocabulary is associated with an occurrences list of document IDs, positions, and TF-IDF weights.

4.5 Module 6: Query Processing and Ranked Retrieval.

The query processing module takes Sindhi text as input, and runs it through the same normalization, tokenization, stop-word elimination and stemming pipeline as the documents, so that query terms and index terms are always in the same canonical stem form (Shah, 2016).

Query Formulation: According to Mahar et al. (2021), a user query Qi is a query set of terms in the vocabulary T=T1,T2,.,Tn and an IR system returns the best subset of documents Ri.

Document Identification: Candidate documents are identified through the inverted index and clustered with the K-means document clustering based on cosine similarity of TF-IDF vectors, which gives good results in text classification with involvement of smaller dataset (Mahar et al., 2021, p. 254). The K-means objective minimized is:

V=âˆ‘_(i=1)^kâ–'âˆ‘_(x_jâˆˆS_i)â–'(x_j-Î¼_i)^2

Pattern Discovery and Topic Modeling: SindhiIRS uses Latent Dirichlet Allocation (LDA) to identify documents based on topic. Every document is a modeled as a multinomial distribution over v topics, with each topic being a multinomial distribution over words: P(w_i|d) = 0.535 + 0.105/10) x P(z j|w i) x P(z j|d) (Mahar et al., 2021). Pattern Discovery (PD) with Apriori algorithm detects common item sets to compare query patterns to document structures (Mahar et al., 2021).

Document Ranking: The retrieved documents are ranked by relevance score: rank(d D) =sum (j=1)v z j sig(z j,d)Q = (z j)V D j z j) where sig(z,d) is the importance of topic z in document d calculated on topical words with a probability stronger than the average (Mahar et al., 2021). This ranking list is sent to the user in descending order of relevance.

5. Implementation

5.1 Graphical User Interface

The automatic Sindhi stemmer and IR system front-end automatically provides a text input-output interface that enables the user to type queries in Sindhi using the Perso-Arabic script and get document results. The interface was tested in Shah (2016) on the stemmer component and further on the entire SindhiIRS to present the ranked document results with relevance scores.

6. Evaluation Methodology

6.1 Test Queries

The part of the 185 male and female students of the Department of Computer Science, Shah Abdul Latif University, Khairpur Mirs, sent queries to the SindhiIRS in three situations (Mahar et al., 2021). Each student created a maximum of 30 queries (a maximum of 10 queries per scenario), which resulted in 5,550 queries in total, where 50 queries per scenario were chosen randomly to be evaluated (Mahar et al., 2021).

The three query situations were:

- Scenario 1: Single-word queries
- Scenario 2: Double-word queries
- Scenario 3: Sentence-length queries

6.2 Evaluation Metrics

SindhiIRS is tested on the two conventional IR metrics (Mahar et al., 2021; Shah, 2016): *Precision=(#(Relevant" " Documents" " Retrieved))/(#(Retrieved" " Documents))* Recall: the percentage of the relevant documents that are recalled:

Recall=(#(Relevant" " Documents" " Retrieved))/(#(Relevant" " Documents)) F-Score: harmonic mean of recall and precision: *F=(2Ã—PrecisionÃ—Recall)/(Precision+Recall)*

In the case of the TPTS preprocessing component, ROUGE evaluation metric (comparing system output to a gold standard) was also used, with an F-score = 0.89, Precision = 0.86, Recall = 0.93 (Nawaz et al., 2023).

6.3 Hardware Configuration

Experiments were performed all on an Intel Core i5 laptop computer with 8 GB RAM and connected to a 10 Mbps LAN and a DSL connection of 2 Mbps (Mahar et al., 2021). Measurement of system efficiency was taken by recording execution time and elapsed time of every query.

7. Results and Discussion

7.1 Scenario 1: Single-Word Query Results.

Metric	***Value***
Minimum precision	*0.33*
Maximum precision	*0.77*
Mean Average Precision (MAP)	*0.6538*
Average documents per query	*2,896*
Number of total documents in database	***7,283.***

Table 4: Scenario 1 findings- single word queries (Mahar et al., 2021)

The single word query MAP of 0.6538 is a good score given the fact that Sindhi has a rich morphology. The lowest accuracy of 0.33 is on queries of highly ambiguous words - a byproduct of the diacritics-omission issue, where one written variant corresponds to multiple words with different semantics (Mahar et al., 2021). The highest of 0.77 is obtained on queries whose root forms are unique and have low frequency. The strong recall provided by morphological stemming is indicated by the average of 2,896 documents retrieved/query (39.7% of the database).

7.2 Scenario 2: Query Results in Double words.

Single word queries are less accurate than the double word queries. Multi-key words queries reduce the number of documents retrieved whereas pattern matching is used to make sure that the two query terms are closely located in the document that was retrieved. According to Mahar et al. (2021), the decisive parameter of setting heuristics and ranking criteria is vicinity and proximity of the keywords (p. 255), and the pattern-matching aspect of SindhiIRS uses the scoring of positional proximity. As it is normal in IR, average precision on double-word queries is higher than the single-word query precision.

7.3 Scenario 3: Sentence-Length Query Results.

The most precise query of the three scenarios is sentence-length queries. SindhiIRS effectively reduces retrieval to the topically most relevant documents by breaking down the query into several stemmed words and ranking the documents based on the similarity of the LDA topic distribution of the query vector to the document. The ranking algorithm rank(d|D) = sum(j=1) v n signific (z j d)Ã—Q (z j)Ã

—V (D j) makes sure that the documents with the most topical content with the query are put on the first page of the results (Mahar et al., 2021).

7.4 Preprocessing Pipeline Accuracy

Component	***Accuracy/Score***
Tokenization	*100% (deterministic rule-based)*
Stop-word removal (TF-IDF)	*522 words found (Nawaz et al., 2023)*
Stemming (SRBS)	*84.85% overall (Shah, 2016)*
POS tagging	*90% (Nawaz et al., 2023; Mahar & Memon, 2010)*
TPTS overall (F-score)	*0.89 vs. baseline 0.81 (Nawaz et al., 2023)*
SindhiIRS MAP (single-word)	***0.6538 (Mahar et al., 2021)***

Table 5: SindhiIRS component and system accuracy.

7.5 Error Analysis

The main causes of retrieval error are:

- Diacritics ambiguity: The absence of vowel diacritics means that the same written word could be based on several roots, and the stemmer would give a single stem to words that are dissimilar (Motlani et al., 2016; Shah, 2016). This is the main reason behind the minimum accuracy of 0.33 (Mahar et al., 2021)
- Boundary errors in compound words: Sindhi compound words that are not written with spaces result in tokenization errors, which are propagated by the stemming and indexing pipeline (Shah, 2016).
- Prefix stemming errors (SER: 25.68%): prefixes of Arabic origin in Sindhi words are the most frequent sources of stemming errors, which testifies to the extent of Arabic borrowing in Sindh (Shah, 2016).
- O-out-of-vocabulary (OOV) words: The 5,327-word SRBS lexicon is just adequate to support a baseline system, but does not exhaust Sindhi vocabulary, especially technical and domain-specific vocabulary in the Computer and Science document categories (Shah, 2016)
- Topic model sparsity: LDA topic models need to have enough training data per topic; smaller document category (Translations: 224 docs, Jail/Housing: 228 docs) can be too sparse to infer any topic (Mahar et al., 2021).

7.6 Comparison with Prior Work

System	**Approach**	***Key Result***
Shah (2016) â€" SRBS	Rule-based stemmer	*84.85% accuracy*
Nawaz et al. (2023) â€" TPTS	Full preprocessing pipeline	*89% F-score*
Mahar et al. (2021) â€" Pattern Discovery IR	LDA + PD	*MAP = 0.6538*
SindhiIRS (this work)	**Integrated IR system (SRBS + TPTS + inverted index + LDA)**	***MAP = 0.6538; preprocessing F-score = 0.89***

Table 6: Comparison with prior Sindhi NLP and IR systems

8. References and Future Work.

SindhiIRS, the first full-fledged automatic Information Retrieval System in Sindhi language, combining a validated NLP preprocessing pipeline with an inverted indexing, TF-IDF weighting of weighting, LDA topic modeling, and document ranking, has been presented in this paper. The system is based on a rule-based stemmer with the accuracy of 84.85% (Shah, 2016), a preprocessing pipeline with the F-score of 89% (Nawaz et al., 2023), and an end-to-end document retrieval system with the mean average precision of 0.6538 on the corpus of 7,283 documents.

The key contributions are:

- The first fully integrated, automatic Sindhi IR system
- A tested preprocessing pipeline that consists of tokenization, 522-word stop word, morphological stemming, and TF-IDF indexing.
- A tri-scenario analysis scheme (single word, double word, sentence queries) that creates a repeatable standard.

Future research that can be determined through the limitations of this study is:

1. Deep learning stemmer: The 38-rule SRBS will be replaced with a neural sequence-to-sequence stemmer trained on the increased STC corpus to lower the prefix SER to 25.68% (Shah, 2016; Sodhar et al., 2023).
2. Diacritics restoration: Adding a diacritics restoration component to disambiguate homographic words before stemming (Sodhar et al., 2023)

3. Named Entity Recognition: Adding the SiNER dataset (Lal et al., 2020) 1.35 million tokens with NER tags to the retrieval pipeline to process entity-aware queries.
4. Expansion of the lexicon through the AMBILE corpus: The AMBILE corpus of more than 2 million cleaned Sindhi tokens (AMBILE, 2025) can be used to expand the lexicon and enhance stemming coverage.
5. Semantic search: Modifying the vector space model to include semantic matching of the Sindhi words (GloVe, Skip-gram, CBOW) trained on the 61-million-word corpus of Narejo and Mahar (2019).

References

Abdul Majid Bhurgri Institute of Language Engineering (AMBILE). (2025). Sindhi language resources now open for AI & NLP innovation. Government of Sindh, Culture, Tourism, Antiquities & Archives Department. Retrieved from https://ambile.pk/sindhi-language-resources-now-open-for-ai-nlp-innovation/

Dootio, M. A., & Wagan, A. I. (2021). Development of Sindhi text corpus. *Journal of King Saud University â€" Computer and Information Sciences*, 33, 468â€"475. https://doi.org/10.1016/j.jksuci.2019.01.006

Lal, M. I., Ul-Mustafa, R., Nawab, R. M. A., Daudpota, S. M., Imran, A. S., & Kastrati, Z. (2020). SiNER: A large dataset for Sindhi named entity recognition. *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, Marseille, France, pp. 2930â€"2938. https://aclanthology.org/2020.lrec-1.361/

Mahar, J. A., & Memon, G. Q. (2010). Rule-based part of speech tagging of Sindhi language. *Proceedings of the 2010 International Conference on Signal Acquisition and Processing*, pp. 101â€"106. IEEE.

Mahar, M. A., Mahar, J. A., Talpur, M. S. H., Mahar, M. A., & Khan, N. (2021). Smart Sindhi documents retrieval system based on pattern discovery approach for students search services. *International Journal of Computational Intelligence in Control*, 13(2), 251â€"261. Retrieved from https://www.mukpublications.com/resources/26.%20Mashooque%20Mahar%20Submission_pagenumber.pdf

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)*, PortoroÅ¾, Slovenia, pp. 2572â€"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*.

Nathani, B., Joshi, N., & Purohit, G. N. (2020). Design and development of unsupervised stemmer for Sindhi language. *Procedia Computer Science*, 167, 1920â€"1927. https://doi.org/10.1016/j.procs.2020.03.212

Nawaz, A., Nawaz, M., Shaikh, N. A., Rajper, S., Baber, J., & Khalid, M. (2023). TPTS: Text pre-processing techniques for Sindhi language. *Pakistan Journal of Emerging Sciences and Technologies (PJEST)*, 4(3). https://doi.org/10.58619/pjest.v4i3.89. Retrieved from https://www.pjest.net/index.php/pjest/article/download/89/29

POS Tagging for Sindhi using Deep Learning. (2024). *Mehran University Research Journal of Engineering and Technology*. Retrieved from https://publications.muet.edu.pk/index.php/muetrj/article/download/2768/827/

Sattar, A. A., Abbasi, S., Rahman, M. U., Baig, A., & Nizamani, M. (2021). Sindhi stemmer using affix removal method. *International Journal of Advanced Trends in Computer Science and*

Engineering, 10(3), 2447â€“2451. Retrieved from https://www.warse.org/IJATCSE/static/pdf/file/ijatcse1331032021.pdf

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.

Shah, M. R., Shaikh, H., Mahar, J. A., & Mahar, S. (2016). Sindhi stemmer for information retrieval system using rule-based stripping approach. *Sindh University Research Journal â€" SURJ (Science Series)*, 48. Retrieved from https://sujo.usindh.edu.pk/index.php/SURJ/article/view/4729

Sodhar, I. N., Hussain, J., Buller, A., & Sodhar, A. (2020). Tokenization of Sindhi text on information retrieval tool. *Pakistan Journal of Emerging Science and Technology*, 1, 10â€"16. Retrieved from https://pjest.com/wp-content/uploads/2021/05/PJEST-004-GICCL-20.pdf

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), 2898â€"2906. https://doi.org/10.17485/IJST/v16i35.1719

Unicode Consortium. (2023). *The Unicode Standard, Version 15.0 â€" Arabic block (U+0600â€"U+06FF)*. Mountain View, CA: Unicode Consortium. Retrieved from https://www.unicode.org/charts/PDF/U0600.pdf

VFAST Transactions on Software Engineering. (n.d.). A systematic approach to probabilistic modeling for retrieving Sindhi text from documents. Retrieved from https://vfast.org/journals/index.php/VTSE/article/view/2010