DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016 *Int. J. Exp. Res. Rev.*, Vol. 43: 214-228 (2024)

Original Article | *Peer Reviewed* | *Open Access*

International Journal of Experimental Research and Review (IJERR)

ISSN: 2455-4855 (Online)
www.iaph.in
ISSN 2455-4855

A Novel Framework for Multilingual Script Detection and Pattern Analysis in Mixed Script Queries

Anu Chaudhary[1*], Rahul Pradhan[2] and Shashi Shekhar[3]

[1]Department of Computer Science and Engineering, GLA University Mathura, India; [2]Department of Computer Science and Engineering, GLA University Mathura, India; [3]Department of Computer Science and Engineering, Amity University, Patna, India

E-mail/Orcid Id:

AC, anu1.deswal@gmail.com, https://orcid.org/0009-0005-4133-5565; *RP,* rahul.pradhan@gla.ac.in, https://orcid.org/0000-0002-5774-4698; *SS,* sshekhar1@ptn.amity.edu, https://orcid.org/0000-0001-8824-1447

Article History:

Received: 04th Aug., 2024
Accepted: 28th Sep., 2024
Published: 30th Sep., 2024

Keywords:

Language identification, Mixed script, Pattern analysis, Script Detection, Word identification

How to cite this Article:

Anu Chaudhary, Rahul Pradhan and Shashi Shekhar (2024). A Novel Framework for Multilingual Script Detection and Pattern Analysis in Mixed Script Queries. *International Journal of Experimental Research and Review*, *43*, 214-228.

DOI:
https://doi.org/10.52756/ijerr.2024.v43spl.016

Abstract: A script detection system that is capable of handling several languages is becoming more necessary in today's world. The task of identifying scripts written in various languages has been substantially facilitated by the use of machine learning and deep learning, respectively. Machine learning techniques have used the Naive Bayes and Support Vector Machines (SVM) mechanism for the purpose of language detection. On the other hand, this paper reviews several unique deep-learning processes that have considered a range of methodologies, including LSTM and Bert. On the other hand, it has been shown that there is a need to improve the accuracy and the scalability often incorporated in multilingual systems. As a consequence of this, the primary focus of the present investigation is on the development of an innovative framework that is capable of recognizing scripts in a variety of languages. In addition, this technique considers pattern analysis while considering mixed script queries. A scalable, efficient, and adaptive approach has been established via study to increase the accuracy of the identification of a large number of languages. Accuracy, recall, and F1-score are some of the performance metrics that have been calculated in order to evaluate the efficacy of the multilingual script identification that has been presented. In conclusion, it has been found that the approach that was provided has supplied a solution that is both efficient and scalable for the detection of multilingual scripts.

Introduction

Information Retrieval is a field of computer science that focuses on satisfying users' information needs through IR systems. As the Internet is increasingly filled with content in languages like Hindi, Marathi, Tamil, and others, the ability to access information in multiple languages has become essential in our globally interconnected society (Shekhar and Sharma, 2020; Ojo et al., 2022; Gupta et al., 2014; Khan and Sawarkar, 2024). The diversity of languages poses a challenge to effective communication in the digital age. Consequently, research in Information Retrieval has gained significant importance in recent years. One of the major challenges in cross-lingual and multilingual information retrieval is obtaining sufficient data when a query is launched in a local language. With the expansion of the World Wide Web, the amount of online content available in languages other than English is increasing. Users would greatly benefit from IR systems that can deliver relevant results in English and local languages.

The spelling of words in text written in an original language but using a different script often deviates from standard rules and instead relies on the pronunciation of the script. Transliteration involves phonetically translating words from a language into a non-native or unfamiliar script (Karmi et al., 2011; Patel and Parikh, 2020; Kumar and Lehal, 2023; Dey et al., 2024). On the internet, the use of the Roman alphabet is growing in popularity for generating content and aiding users in finding information. Before applying other natural language processing (NLP)

*Corresponding Author: anu1.deswal@gmail.com

techniques, the data needs to undergo pre-processing, which may include translation and/or transliteration. Transliteration serves as a means for machine translation (MT) and cross-lingual information retrieval (CLIR).

Transliteration can be approached in two different ways. The first method is forward transliteration, which occurs when native words are written in an alien or foreign script. For example, the Hindi term जीवन (written in Devanagari script) translates to "life" in English and can be transliterated as jivan, Jeevan, jeeivan, or various other versions. On the other hand, back-transliteration involves translating a word from a non-native script back to its original script. In this case, "Jivan" would be back-transliterated to its original Devanagari script. While back-transliteration requires producing the same original word, forward transliteration offers more creative freedom to the transliterator. Karimi et al. (2011) conducted extensive research, but their seminal piece still summarizes machine transliteration well. In recent years, multilingual social media posts have increased, making it harder for IR systems to process and retrieve pertinent texts.

Related Work

Various natural language processing (NLP) applications, including code-mixed language classification, have been addressed and improved using a variety of ML methods and neural networks. When two or more languages' vocabulary and syntax are mixed together in a single sentence, this is called "code mixing," according to Sristy et al. (2017), Feurer and Hutter (2019), Chaitanya et al. (2018). Code mixing is also used when two languages are spoken at the same time. Code mixing occurs most often in casual circumstances, reflecting the conversants' propensity to switch languages while communicating, and it is clear that both languages are used concurrently in all grammatical and lexical components. Shekhar et al. (2020), Thara and Poornachandran (2018) and Patel and Bhattacharyy (2019) proposed a method for determining the language of bilingual text that was presented using Facebook, Twitter, and WhatsApp datasets. Some quantum LSTM network subclasses proficiently learned and predicted language in social media material. Regardless of the exact Hamiltonian form, the results show that ML techniques have a lot of room to grow in quantum dynamics.

An extensive experiment using transfer learning and fine-tuning of BERT models was carried out by (Ansari et al., 2021) to decipher the language used in Twitter data. This study used a dataset that included code-mixed texts in Hindi, English, and Urdu for pre-training and word-level language classification processes. Pre-trained code-mixed representations outperform monolingual ones.

The primary emphasis is identifying mixed scripts within a dataset that include Roman Urdu, Hindi, Saraiki, Bengali and English (Yasir et al., 2021; Naosekpam and Sahu, 2023). In order to train the language identification model, the researchers utilised RNN and word vectorisation approaches. Moreover, they enhanced numerous model structures, including BGRU, GRU, bidirectional LSTM (Sasidhar et al., 2020; Anand et al., 2022) and long short-term memory. The study attained a high-performance score through experimentation. Roman-English word-styling, generative spellings, and phonetic typing are only a few of the multilingual difficulties explored in the study.

The document's language was successfully deciphered word-by-word in code-mixed English, Bodo Assamese, and other languages (Mosa, 2020; Ojo et al., 2022). In order to analyse and predict the language of Facebook-sourced content, the researchers used a variety of categorisation approaches. The models' word-level language detection accuracy varied because they were trained on the code-mixed corpus utilising features based on n-grams and dictionaries. Building upon Conditional Random Fields (CRF), the method demonstrated in allows for word-level language detection in code-mixed text (Thara and Poornachandran, 2018). This method relies on lexical, contextual, character n-gram, and unique character properties, making it applicable to a wide range of languages. Across a variety of language pairs, the experimental results show that the CRF-based method outperforms alternative datasets time and time again. Researchers used datasets of chat conversations written in a combination of English-Bengali and English-Hindi to identify word-by-word language transitions (Dutta et al., 2015). The author evaluated the system's performance in several languages and created a code-mixing index to measure the amount of language blending in the corpora. Standard transliterations sometimes include the interchange of certain characters, and Sarma et al. (2018) presented various ways to learn this sequence. Using the given transliterations as examples, the researchers demonstrated how these algorithms outperformed competing methods in identifying Hindi words. Their one-of-a-kind experimental model considers language along with part-of-speech of nearby words while attempting to identify languages at the word level. Experimental findings clearly show that the proposed model achieves better accuracy than prior methods. An approach to the problem of syllable along with character n-gram identification in code-mixed and multi-script texts, was

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

proposed by Shashirekha et al. (2022) to improve ML classifiers. We tested the suggested models with three Dravidian language pairs: Malayalam and English, Tamil and English, Kannada and English. ML classifiers' output showed that code-mixed and multi-script texts might be better analysed with the addition of syllables along with character n-gram features.

In order to identify words in code-mixed data at the language level, Mandal and Singh (2018) developed a novel framework for language tagging using a multichannel neural network that integrates CNN with LSTM (Shekhar et al., 2018; Jitta et al., 2017; Kozhirbayev et al., 2018; Shanmugalingam et al., 2018; Velankar et al., 2022). The multichannel neural network showed good results in language identification when combined with a Bi-LSTM-CRF context capture module, thanks to this architecture's integration of contextual information.

According to the above literature, machine transliteration systems encounter several challenges, including:

1. Script Requirements: Determining the appropriate script for transliterating a particular word or name can be complex, especially when dealing with multilingual texts where multiple scripts may be used.
2. Sound Gaps: Some languages may have sounds that do not exist in the target language, leading to difficulties in phonetic representation during transliteration.
3. Transliteration Variations: Different transliteration variations may exist for the same word or name, resulting in inconsistencies in the transliteration process.
4. Language of Origin: Identifying the language of origin of a word or name is crucial for accurate transliteration. However, in code-mixed or multilingual texts, this task can be challenging.

Table 1. Summary of state of the art Model/Approaches.

Year	Author	MT Model / Approach	Strength	Research Gap
2022	Velankar et al., 2022	DL based approaches, naïve bayes, SVM	Identifying and categorising hate speech on Twitter and Facebook databases	Opinions on certain subjects shift across time
2022	Chakravarthi et al., 2022	Machine learning and deep learning	The language used by David in the coded sample	Limited resource dataset for other Dravidian languages.
2021	Ravikiran and Annamalai, 2021	Mulitlingual BERT and Distil BERT Model used	Database for (DOSA) used for code-mixed text in Tamil and English.	Findings from less complex models, such as LSTM-CRF and its derivatives, are omitted.
2020	Shekhar et al., 2020	BiLSTM	Determine the programming language Contradictory information	Performing an analysis of brief textual material
2019	Shashirekha et al., 2022	Machine learning	Recognise Hate Speech and Detect Offensive Language	Discontinued in mixcode
2018	Sharma and Mittal, 2018	OOVTTM model	Improving word combinations found in dictionaries	Words pertaining to named objects have been translated incorrectly.
2016	Palangi et al., 2016	RNN Learning	Useful for words with semantic meaning	Proficiency with the subject area's lexicon is necessary
2015	Raghavi et al., 2015	SVM	Sorting social forum topics into categories based on their languages	Normalisation of term variation in code-mixed data exists
2015	Roy et al., 2015	Grapheme-cooccurrence, corpus Matching for MLM	Intent word detection	Not well-suited for multiple-word
2014	Gella et al., 2014	Word identification, SVM	Transliterate or Translate	Problems with transliteration, Badha, Badhaa, Barha, and others.

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

5. Translation vs. Transliteration: Deciding whether a name or word should be translated or transliterated (or a combination of both) requires careful consideration and context-aware processing.

Addressing these difficulties is essential for improving accuracy and machine transliteration systems' effectiveness.

Proposed Framework for language detection and pattern Analysis in Mixed Script Queries

As shown in figure 1, the proposed framework identifies the language from mix-code text, processes it and returns the intention of that query or sentence.

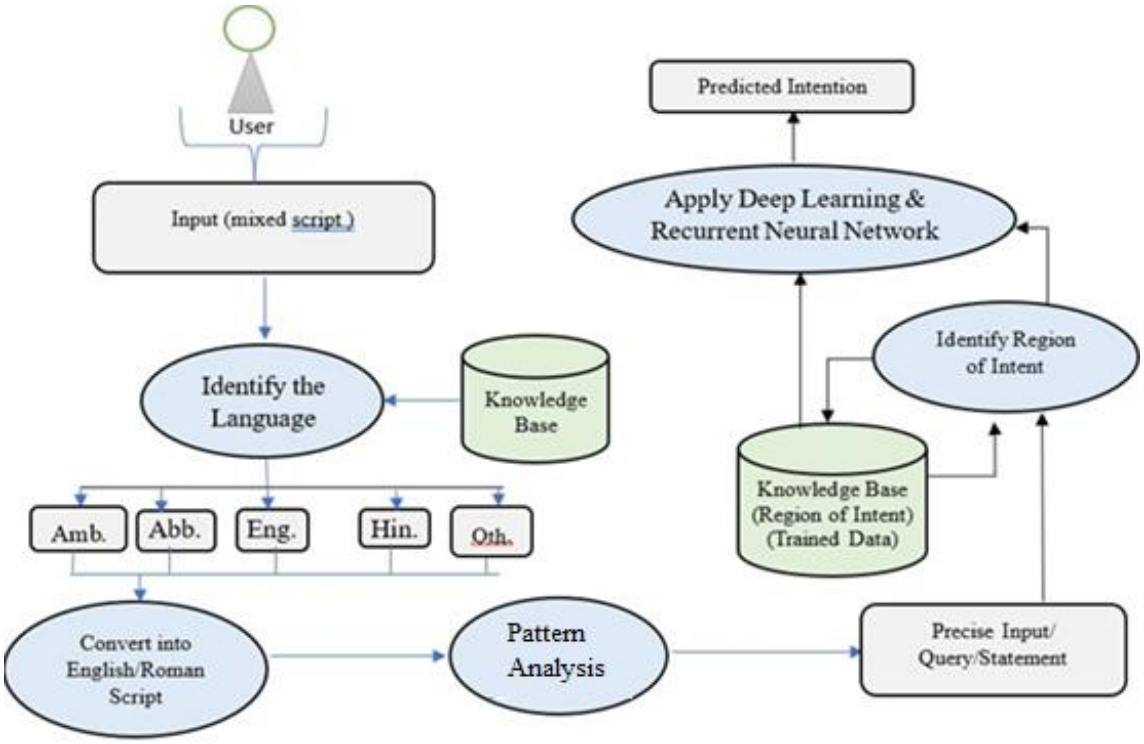

Figure 1. A Framework for Pattern Analysis and Intent Identification in Mixed Script Queries.

In this model, the user submits mix-code scripts/sentences, and the language identifier finds all keywords/tokens of user scripts/sentences in their language with the help of a knowledge base and converts them into English/Roman script. After the preprocessing model, put this script in a particular region of intent with the help of a knowledge base (trained dataset) and apply deep learning and RNN to predict intention.

Language Identification

The script that users enter could be mixed code or multilingual. Create a label sequence using word-level classification and use Bidirectional LSTM (Kazi et al., 2020; Mandl et al.. ,2020; Mabokela, 2019) for sentence-level classification. But looking at it from a problem-solving standpoint, it's all in Roman letters. In order to train a word-level classifier to understand both native and English terms, we employ words from each of the languages, including Hindi, English, and knowledge-based. The sentence-level language identification method is exclusive to the English language. To improve the word-level classifier's accuracy, the labelling sequence is utilised. When the classifier is confused about the meaning of a word, this usually helps with the labelling process. Mislabeling occurs due to overlap between the two languages' shorter words. In such a case, the label of the word that comes before or after it can provide useful context for understanding the meaning of the term in question. Language identification procedure is depicted in figure 2, which is provided below.

Algorithm: Procedure Language Identification ()
{
Input: mixCodeScript (string) ***//Read mix-code script/sentence/query from the user***
vectorSequence = BiDirectionalLSTM(mixCodeScript)
//Generate sequence of vectors using Bi-Directional LSTM
sentenceClassification = SentenceLevelClassification(vectorSequence)
//Apply sentence level classification
wordClassificationInput = PrepareWordClassificationInput(vectorSequence, sentenceClassification)
// Forward to word level classification process
labeledSequence = WordLevelClassification (wordClassificationInput, knowledgeBase)
// Apply word level classification with knowledge base

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

unilingualOutput = Transliterate(labeledSequence)
// Transliterate the output of word level classification into unilingual (English/Roman)
Output(unilingualOutput) ***// Output the result***
}

(Hindi, English, and Other), and if the list of other words is not empty, it undergoes the same process once more. This involves converting all possible words into Hindi or English lists using KB_ABB and the knowledge base. Furthermore, context analysis is employed for ambiguous

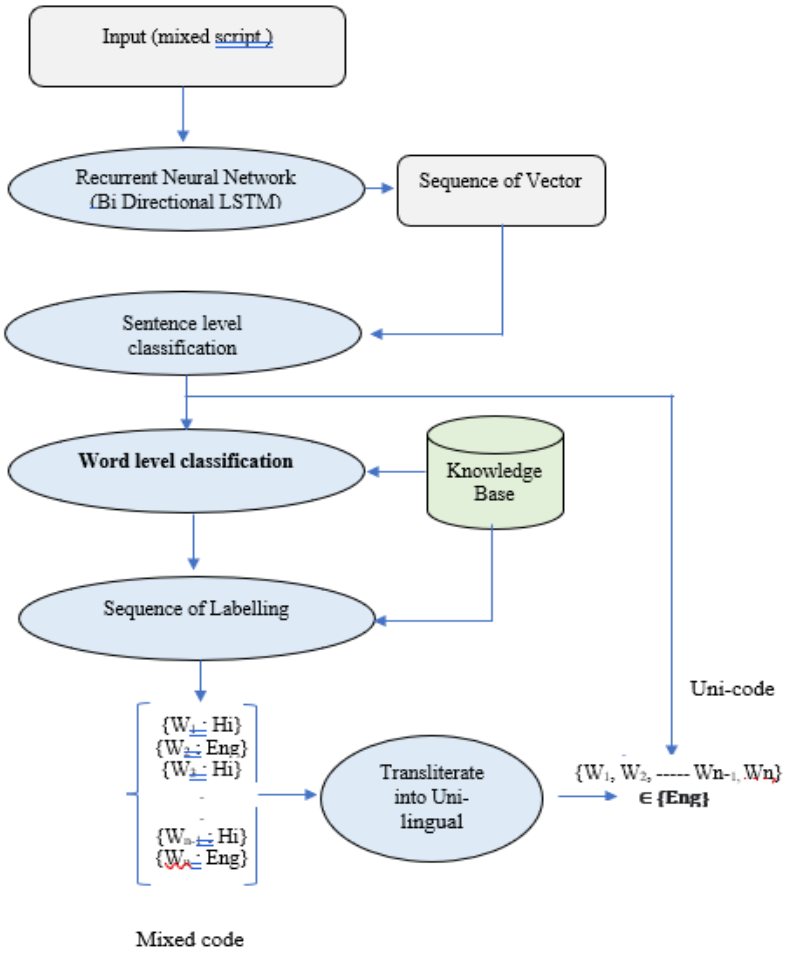

Figure 2. Flow diagram for language identification.

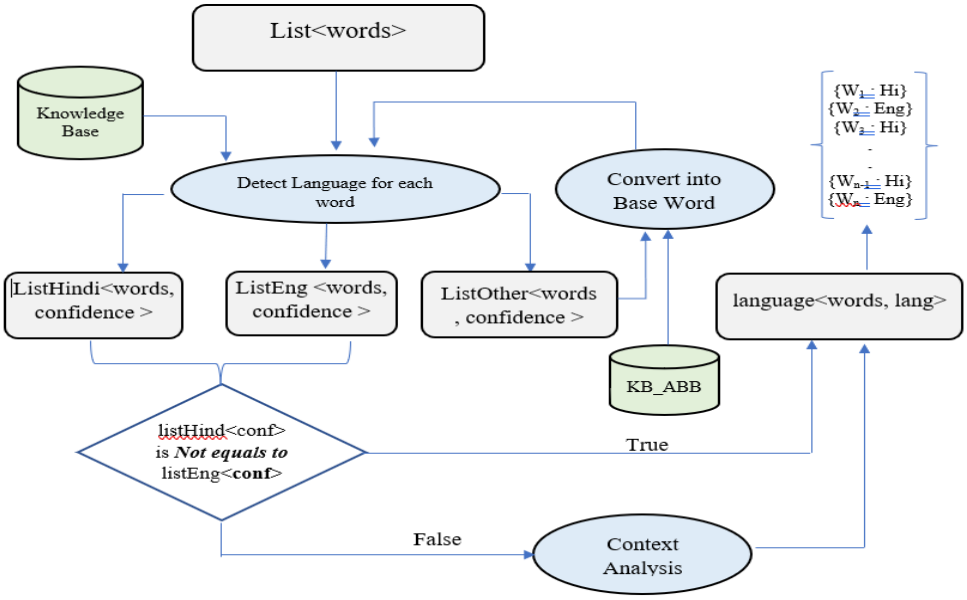

Figure 3. Process to classify the words.

Figure 3 illustrates the intricate process of word classification. The words are categorized into three lists words, resulting in a set of words paired with their corresponding languages.

The structure of KB_ABB, as shown in Table 2, supports the word classifier to train the system to understand the abbreviations/short-length words.

Table 2. Representation of KB_ABB for abbreviations.

English/Roman Script	List of Abbreviations Roman words
FINE	F9->5N->FYN
GREAT	gr8
FINE BY ME	FBN
For Your Information	FYI
To be Honest	TBH
Did you know	DYK
By the Way	BTW
As soon as Possible	ASAP
Oh My God	OMG
NI8	Night
See	C
Brother	BRO
Be	B
Before	B4
Best Friend Forever	BFF
End of Day	EOD
See you tomorrow	CYT
Are	R
You	U
Am	M

The structure of the Knowledge Base shown in table 3 supports the word classifier to train the system to understand the native words.

Table 3. Hash table Representation of Knowledge Base.

Base Words	Similar possibilities	Base Words	Similar possibilities
Tera	Teraa	khushboo	khushbuu
	Thera		khushbu
	Teraaa		khusbhu
mujhe	muze	men	mein
	mujhse		meein
	mujeh		maein
	mujh		menu
	muhje		main
	mujhey		
	muzhe		
	muhjhe		
	mujkhe		

Algorithm: Word Level Classification:

Word_level_language_detection(mixed_code)
{
STEP -1:*split mixed-code into words*
List<words>words= tokenization(mixed-code)
STEP -2:***Find out confidence level of each word with different vocabulary***
For All words of list
If (word$_i$ **∈ {**knowladgeBase**,** engDictionary**}**
List<word$_i$, confidence, language>
listHindi=detectLanguage (word$_i$, knowladgeBase);
List<word$_i$, confidence, language>
listEng=detectLanguage (word$_i$, engDictionary);
Else
List<word$_i$, confidence, language>listOther=detectLanguage (word$_i$, engDictionary);
End Loop
*If (*listOther is notEmpty)
update (List<words>words, KB_ABB);
Repeat step 2;
STEP -3:***Consider final language of word with max confidence value***
For all listHindi and listEng
if(listHindi <confidence> != listEng<confidence >)
List<word, language> language = max(listHindi<confidence >, listEng<confidence >);
Else
// words which have confusion, need to apply context Analysis
List<word, language> language= contextAnalysis(listHindi, listEng)
End if
End Loop}

Experimental Evaluation

The authors shared their experiments' findings on a dataset containing various mixed scripts used by users on different social media platforms. Every word in this dataset has been tagged with one of two languages: mixed code and numbers, digits, and special symbols. The text was culled from social media. Twenty scripts should be considered for classification after preprocessing. Table 3's first column gives the script, and the second column shows the number of sentences or scripts. Tables 4 and 5 summarise the sample dataset that has been annotated at sentence level. The sentence-level annotations are included in Table 5, together with the word-level annotations that were obtained from them.

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

Table 4. Sample dataset: The scripts taken from various social media app/data sets taken from MSIR.

Sample Dataset (scripts are used on various social media app)	
1	main aaj main market jaunga
2	Tum bahut dust ho
3	Tum sab log aajao
4	Aaj main khush hu becoz today is my birthday
5	BTW main kal aa jayuga
6	How r u
7	I m f9
8	Tere Suit Ke Re Saare Re Colour Baawali Tere Aage Saari Chhori Sai Blur Baawali
9	Today is my Birthday. Or Mai Bahut hee khush hun
10	Kya tum is restaurant main ek table book karne main meri help karoge
11	Taj Mahal is in India. Ye Bahut hi khoobsurat hai
12	Log Bol rhe hai Jaishah ne world cup ki team khareed le hai
13	Code deploy hone may abhi time lagega
14	University ne abhi students ki marksheet nhi send ki hai
15	Mera resume abhi updated nhi hai.
16	Aajkal sabi Paytm use kar rhe hai.
17	Mujhe Bank may paise deposit karne hai.
18	Morning may sabhi ko walk karni chahiye.
19	Hello may bol rhi hu How r u.
20	Teacher ne sabhi Topic cover karwa diye hai.

Performance of the proposed system is measured through precision, recall, accuracy, and f-measure. Precision, recall, accuracy, and F-score are defined in equations 1, 2, 3 and 4, respectively. Accuracy is a very important performance metric, which is the result of the ratio of predicted (TP+TN) to all (TP+TN+FP+FN) observations, as given in equation (1).

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

Precision (P) *is ratio of relevant (TP) and all retrieved (TP and FP) words as* given in equation (2)

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

Recall (R) The proportion of recovered and relevant words to all relevant languages available is indicated by the recall (R) as given in equation (3).

$$\text{Recall (R)} = \frac{TP}{TP + FN} \quad (3)$$

F-Measure is the average value of Precision or Recall weights as given in equation (4). False positive and negative both values are considered as a result.

$$\text{F Measure} = \frac{2 * (precision * Recall)}{(Precision + Recall)} \quad (4)$$

Table 5. Description of sample dataset annotated at sentence level.

Language	#Sentences	Avg length
Hindi (hn)	3	9 words
English (en)	2	3 words
MixedCode-(mc)	15	9 words
Total	**20**	

Table 6. Explanation of word-level annotations acquired through sentence-level annotations.

Status	Language	Total Words
Resolved	Hindi (hn)	61
	English (en)	49
Unresolved		44
	Total	154

Using algorithm 1 and algorithm 2, each word language is identified in the script for a dataset as given in Table-3.

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

Table 7. Confidence level/Probability of each word with different vocabulary.

Sentence ID	Word	Lang. Detected	Probability
1	main	Amb	1.00000
	Aaj	Hi	1.00000
	main	Amb	1.00000
	Market	En	0.99804
	Jaunga	Hi	1.00000
2	Tum	Hi	0.99999
	Bahut	Hi	1.00000
	Dust	Amb	0.71428
	Ho	Hi	0.71428
3	Tum	Hi	0.99999
	Sab	Hi	1.00000
	Log	En	0.99999
	Aajao	Hi	1.00000
4	Aaj	Hi	1.00000
	Main	Amb	0.99999
	Khush	Hi	1.00000
	hu	Hi	0.85714
	becoz	En	1.00000
	Today	En	1.00000
	Is	Oth	0.99999
	my	En	0.71428
	birthday	En	1.00000
5	BTW	Abb	1.00000
	Main	Amb	1.00000
	Kal	Hi	0.85714
	aa	Hi	1.00000
	Jayuga	Hi	0.99999
6	How	En	0.71428
	R	Abb	1.00000
	U	Abb	0.99999
7	I	En	0.99999
	M	Abb	1.00000
	f9	Abb	0.85768
8	Tere	Hi	0.99999
	Suit	Hi	1.00000
	Ke	Oth	0.99999
	Re	Hi	1.00000
	Saare	Hi	1.00000
	Re	Hi	0.99999
	Colour	En	0.71777
	Baawali	Hi	1.00000
	Tere	Hi	0.99999
	Aage	Hi	1.00000
	Saari	Hi	0.99999
	Chhori	En	1.00000
	Sai	Amb	0.85714
	Blur	En	0.85714
	Baawali	Hi	1.00000

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

9	Today	Hi	1.00000
	is	Oth	0.99999
	my	En	0.85714
	Birthday	En	0.85714
	Or	Amb	1.00000
	Mai	Hi	1.00000
	Bahut	Hi	1.00000
	hee	Hi	1.00000
	khush	Hi	0.85714
	Hun	Hi	0.99999
10	Kya	Hi	1.00000
	tum	Hi	1.00000
	Oth	Hi	0.99999
	restaurant	En	0.85558
	main	Amb	0.99999
	ek	Oth	0.99999
	table	En	0.85714
	Book	En	1.00000
	Karne	Hi	0.99999
	Main	Amb	1.00000
	Meri	Hi	0.85684
	Help	En	1.00000
	Karoge	Hi	0.85714
11	Taj	Amb	1.00000
	Mahal	Hi	1.00000
	is	Oth	0.99999
	in	Amb	0.99999
	India	En	1.00000
	Ye	Hi	1.00000
	Bahut	Hi	1.00000
	hi	Amb	0.99999
	khoobsurat	Hi	1.00000
	Hai	Hi	0.46875
12	Log	En	0.98956
	Bol	Oth	0.64093
	Rhe	Hi	0.98828
	Hai	Hi	0.46875
	Jaishah	Hi	0.80107
	Ne	Oth	0.21705
	World	En	1.00000
	Cup	En	1.00000
	Ki	Oth	0.55078
	Team	En	1.00000
	Khareed	Hi	1.00000
	Le	Oth	0.35573
	hai	Hi	0.46875
13	Code	En	1.00000
	Deploy	En	0.65819
	Hone	Amb	0.87206
	May	Amb	0.98047
	Abhi	Amb	0.78506
	Time	En	1.00000
	Lagega	Hi	1.00000
14	University	en	1.00000

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

	Ne	Oth	0.21705
	Abhi	Amb	0.78506
	students	En	0.98654
	Ki	Oth	0.55078
	marksheet	Hi	0.40933
	Nhi	Hi	0.41016
	Send	En	1.00000
	Ki	Oth	0.55078
	Hai	Hi	0.46875
15	Mera	En	0.70000
	Resume	En	1.00000
	Abhi	Amb	0.78506
	updated	en	1.00000
	Nhi	Hi	0.41016
	Hai	Hi	0.46875
16	Aajkal	Hi	0.89960
	Sabi	Oth	0.98047
	Paytm	En	0.94271
	Use	en	1.00000
	Kar	Oth	0.52713
	Rhe	Hi	0.98828
	Hai	Hi	0.46875
17	Mujhe	Hi	1.00000
	Bank	En	0.94282
	May	En	0.98047
	Paise	En	0.76802
	deposit	En	0.98949
	Karne	Oth	0.58984
	Hai	Hi	0.46875
18	Morning	En	1.00000
	May	En	0.98047
	Sabhi	Hi	1.00000
	Ko	Oth	0.35433
	Walk	En	1.00000
	Karni	En	0.71354
	chahiye	Hi	0.98764
19	Hello	En	1.00000
	May	En	0.98047
	Bol	Oth	0.64093
	Rhi	En	0.79346
	Hu	En	0.75196
	How	En	1.00000
	R	Abb	0.93519
	U	Abb	0.76133
20	Teacher	En	1.00000
	ne	Oth	0.21705
	sabhi	Hi	1.00000
	Topic	En	1.00000
	cover	En	1.00000
	karwa	Oth	0.91016
	diye	Oth	0.84942
	hai	Hi	0.46875

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

Word-level classification is experiencing inaccuracies, even though the identification of mixed languages is accurate. When dealing with shorter words, the surrounding word labels play a crucial role in determining the language of the current word. In this system, KB_ABB and the knowledge base assist in determining words in specified languages. Calculate the frequencies of Hindi, English, other, Ambiguous (Amb) and abbreviation (Abb) words in a script as a proportion of the total words.

The summarized word-level results for the aforementioned categories are presented in Table 7, and a graphical representation of the summary can be found in Figure 4. In the proposed system described above, all words identified as abbreviations (Abb) utilize KB_ABB, while ambiguous (Amb) words are detected using a Knowledge Base designed to return words based on user context. Evaluate Precision, Recall, F-Measure, and Accuracy for word-level identification using equations 1,

Table 8. Sentence/script wise Words- Level Identification.

Sentence ID	Total Word	Hindi	English	Accuracy	Others	Amb	Abb
1	5	2	1	0.6	0	2	0
2	4	3	0	0.8	0	1	
3	4	3	1	1.0	0	0	0
4	9	3	4	0.8	1	1	0
5	5	3	0	0.6	0	2	0
6	3	0	1	0.3	0	0	2
7	3	0	1	0.3	0	0	2
8	15	10	3	0.9	1	1	0
9	10	6	2	0.8	1	1	0
10	13	6	4	0.8	1	2	0
11	10	5	1	0.6	1	3	0
12	13	5	4	0.7	4	0	0
13	7	1	3	0.6	0	3	0
14	10	2	4	0.6	3	1	0
15	6	2	3	0.8	0	1	0
16	7	3	2	0.7	2	0	0
17	7	2	4	0.9	1	0	0
18	7	2	4	0.9	1	0	0
19	8	1	4	0.6	1	0	2
20	8	2	3	0.6	3	0	0

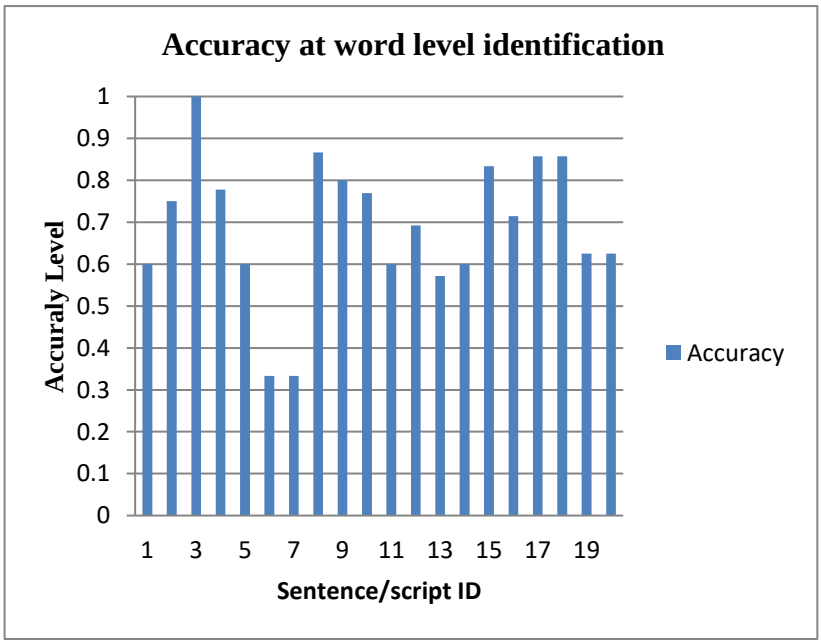

Figure 4. Accuracy at word level identification.

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

Table 9. Precision, Recall, F-Measure and Accuracy at world level identification.

Sen_ID	Precision	Recall	Accuracy	F-measure
1	0.60000	1.00000	0.71429	0.75000
2	0.75000	1.00000	0.80000	0.85714
3	1.00000	1.00000	1.00000	1.00000
4	0.77778	0.87500	0.72727	0.82353
5	0.60000	1.00000	0.71429	0.75000
6	0.33333	1.00000	0.60000	0.50000
7	0.33333	1.00000	0.60000	0.50000
8	0.86667	0.92857	0.82353	0.89655
9	0.80000	0.88889	0.75000	0.84211
10	0.76923	0.90909	0.75000	0.83333
11	0.60000	0.85714	0.64286	0.70588
12	0.69231	0.69231	0.52941	0.69231
13	0.57143	1.00000	0.70000	0.72727
14	0.60000	0.66667	0.50000	0.63158
15	0.83333	1.00000	0.85714	0.90909
16	0.71429	0.71429	0.55556	0.71429
17	0.85714	0.85714	0.75000	0.85714
18	0.85714	0.85714	0.75000	0.85714
19	0.62500	0.83333	0.63636	0.71429
20	0.62500	0.62500	0.45455	0.62500

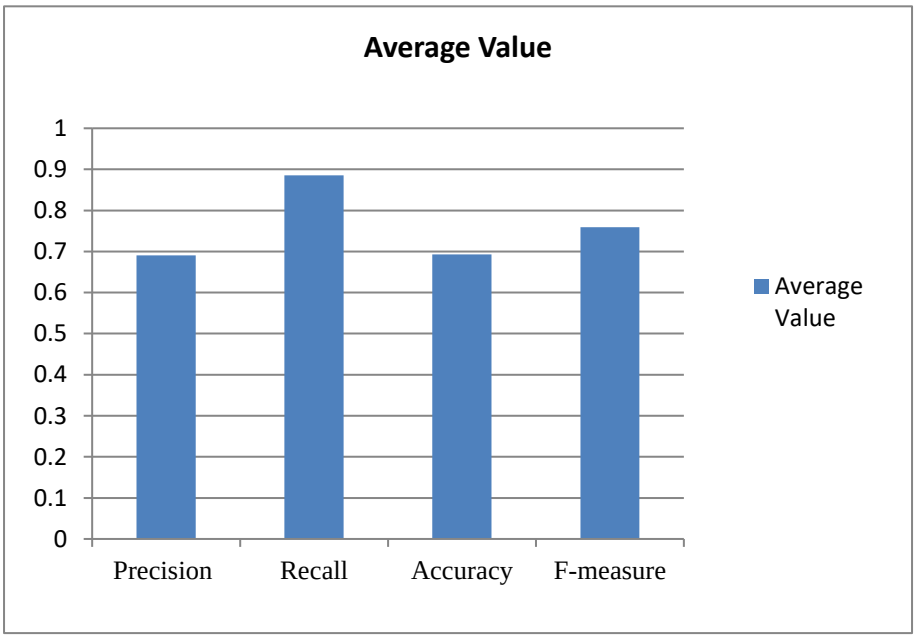

Figure 5. Average value of Precision, Recall, F-Measure and Accuracy.

2, 3 and 4. The outcomes are detailed in Table 8, and a graphical representation is depicted in Figure 5.

To understand the overall performance of the proposed system, we summarize average precision, recall, accuracy, and F-score. The average summarized details are shown in figure 5. Average F-score for the proposed system is 0.7559 and the accuracy is 0.6927. Hence, the proposed system performs better.

Conclusion

In the realm of linguistic diversity, the integration of machine learning techniques has yielded a remarkable framework capable of adeptly parsing mixed-script text, thereby introducing an innovative approach to language

DOI: https://doi.org/10.52756/ijerr.2024.v43spl.016

detection (Kazi et al., 2020). This research stands out for its emphasis on the intricate dynamics of script amalgamation, particularly within the context of Hindi-English bilingual users on various social media platforms. The framework's excellence lies in its utilization of sequence-to-sequence models and attention mechanisms for pattern analysis, showcasing superior accuracy in both language detection and pattern extraction capabilities. Within the proposed system, the identification of abbreviations (Abb) leverages KB_ABB, while ambiguous (Amb) words are discerned through a Knowledge Base designed to adapt to user context. Word-level identification is made to thoroughly comprehend the system's performance using accuracy, precision, recall, and F-measure evaluation metrics. A varied dataset is used to conduct experiments, combining scripts common in social media user-generated material. Text from various social media sources is included in this dataset, and every word has been painstakingly labelled with one of two languages, which include a combination of mixed code, numbers, figures, and unique symbols. Using twenty different scripts for analysis, preprocessing techniques are used to get the dataset ready for categorisation. Tucked away in Table-3 are the findings that capture the spirit of the sentence/script numbers and the script descriptions that go along with them. Tabulated in Table 4, the sample dataset annotations are further broken down to the phrase level for easy understanding. In order to further comprehend the complexities of the dataset, Table 5 offers insights into word-level annotations that are generated from the basic sentence-level annotations. The suggested system's performance is summarised in Figure 6, which also includes the average recall, accuracy, and F-score. Importantly, the study achieves a remarkable 0.6927 accuracy and an average F-score of 0.7559. When applied to the problems of mixed-script text analysis in the ever-changing social media environment, these findings demonstrate how well the suggested method performs. With its ability to handle complex script interactions and different language patterns, the framework is a major step forward in the ever-changing field of language processing. This is especially true in today's world of highly connected and multilingual cultures.

Conflict of interest

Nil

References

Anand, M., Sahay, K.B., Ahmed, M.A., Sultan, D., Chandan, R.R., & Singh, B. (2022). Deep learning and natural language processing in computation for offensive language detection in online social networks by feature selection and ensemble classification techniques. Theor. *Comput. Sci., 943*, 203-218.

Ansari, M. Z., Beg, M. S., Ahmad, T., Khan, M. J., & Wasim, G. (2021). Language Identification of Hindi-English tweets using code-mixed BERT. IEEE, In *2021 IEEE 20th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, pp. 248-252. https://doi.org/10.1109/ICCICC53683.2021.9811292.

Chaitanya, I., Madapakula, I., Gupta, S. K., & Thara, S. (2018). Word level language identification in code-mixed data using word embedding methods for Indian languages. IEEE. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1137-1141. https://doi.org/10.1109/ICACCI.2018.8554501.

Chakravarthi, B. R., Priyadharshini, R., Muralidaran, V., Jose, N., Suryawanshi, S., Sherly, E., & McCrae, J. P. (2022). Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *Language Resources and Evaluation*, *56*(3), 765-806. https://doi.org/10.1007/s10579-022-09583-7.

Dey, S., Thakur, S., Kandwal, A., Kumar, R., Dasgupta, S., & Roy, P.P. (2024). BharatBhasaNet-A Unified Framework to Identify Indian Code Mix Languages. *IEEE Acess, 12*, 68893-68904. https://doi.org/10.1109/ACCESS.2024.3396290

Dutta, S., Saha, T., Banerjee, S., & Naskar, S. K. (2015). Text normalization in code-mixed social media text. IEEE, In *2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS)*, pp. 378-382. https://doi.org/10.1109/ReTIS.2015.7232908.

Feurer, M., & Hutter, F. (2019). Hyperparameter optimization. *Automated machine learning:* for multi-script information retrieval. ©TheAuthor(s) 2019 F. Hutter et al. (eds.), Automated Machine Learning, The Springer Series on Challenges in Machine Learning, pp. 1-33. https://doi.org/10.1007/978-3-030-05318-5_1

Gella, S., Bali, K., & Choudhury, M. (2014). ye word kis lang ka hai bhai? Testing the Limits of Word level Language Identification. In *Proceedings of the 11th International Conference on Natural Language Processing*, pp. 368-377.

Gupta, P., Bali, K., Banchs, R. E., Choudhury, M., & Rosso, P. (2014). Query expansion for mixed-script information retrieval. SIGIR '14: Proceedings of the

37th international ACM SIGIR conference on Research & development in information retrieval. pp. 677 – 686. https://doi.org/10.1145/2600428.2609622

Jitta, D. S., Chandu, K. R., Pamidipalli, H., & Mamidi, R. (2017). nee intention enti? towards dialog act recognition in code-mixed conversations. *IEEE*, In *2017 International Conference on Asian Language Processing (IALP)*, pp. 243-246.

Karimi, S., Scholer, F., & Turpin, A. (2011). Machine transliteration survey. *ACM Computing Surveys (CSUR)*, *43*(3), 1-46. https://doi.org/10.1145/1922649.1922654.

Kazi, M., Mehta, H., & Bharti, S. (2020). Sentence level language identification in Gujarati-Hindi code-mixed scripts. IEEE, In *2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC)*, pp. 1-6. https://doi.org/10.1109/iSSSC50941.2020.9358837

Khan, Z. F., & Sawarkar, S.D. (2024). Enhancing Sentiment Analysis of Marathi-English Code-Mixed Texts using an Ensemble Model. *International Journal of Intelligent Systems and Applications in Engineering*, *12*(18s), 741. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/5038

Kozhirbayev, Z., Yessenbayev, Z., & Makazhanov, A. (2018). Document and word-level language identification for noisy user generated text. IEEE, In *2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*, pp. 1-4. https://doi.org/10.1109/ICAICT.2018.8747138.

Kumar, A., & Lehal, G. S. (2023). A Hybrid Approach for Complex Layout Detection of Newspapers in Gurumukhi Script Using Deep Learning. *International Journal of Experimental Research and Review*, *35*, 34–42. https://doi.org/10.52756/ijerr.2023.v35spl.004

Mabokela, K. R. (2019). A multilingual ASR of Sepedi-English code-switched speech for automatic language identification. IEEE, In *2019 International Multidisciplinary Information Technology and Engineering Conference (IMITEC)*, pp. 1-8.

Mandal, S., & Singh, A. K. (2018). Language identification in code-mixed data using multichannel neural networks and context capture. *arXiv preprint arXiv:1808.07118*. dhttps://doi.org/10.18653/v1/w18-6116.

Mandl, T., Modha, S., Kumar M, A., & Chakravarthi, B. R. (2020). Overview of the havoc track at Fire 2020: Hate speech and offensive language identification in Tamil, Malayalam, Hindi, English and German. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, pp. 29-32.

Mosa, M. A. (2020). A novel hybrid particle swarm optimization and gravitational search algorithm for multi-objective optimization of text mining. *Applied Soft Computing*, *90*, 106189. https://doi.org/10.1016/j.asoc.2020.106189.

Naosekpam, V., & Sahu, N. (2023). A Hybrid Scene Text Script Identification Network for Regional Indian Languages. A*CM Trans. Asian Low-Resour. Lang. Inf. Process.*, *23*(8), Article 124 (26 pages). https://doi.org/10.1145/3649439

Nayel, H. A., & Shashirekha, H. L. (2019). DEEP at HASOC2019: A Machine Learning Framework for Hate Speech and Offensive Language Detection. In *FIRE (working notes)*, pp. 336-343.

Ojo, O.E., Gelbukh, A., Calvo, H., Feldman, A., Adebanji, O.O., & Armenta-Segura, J. (2022). Language Identification at the Word Level in Code-Mixed Texts Using Character Sequence and Word Embedding. *Proc. 19th Int. Conf. Nat. Lang. Process. Shar. Task Word Lev. Lang. Identif. Code-mixed Kannada-English Texts*, pp. 1–6, 2022.

Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., ... & Ward, R. (2016). Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *24*(4), 694-707. https://doi.org/10.1109/TASLP.2016.2520371.

Patel, D., & Parikh, R. (2020). Language Identification and Translation of English and Gujarati code-mixed data. IEEE, In *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1-4. https://doi.org/10.1109/ic-ETITE47903.2020.410.

Patel, P., & Bhattacharyya, P. (2019). Recent Work in Machine Transliteration for Indian Languages, pp. 1-12.

Prabhakar, D.K., & Pal, S. (2018). Machine transliteration and transliterated text retrieval: a survey. *Sādhanā, 43*, 93. https://doi.org/10.1007/s12046-018-0828-8

Raghavi, K. C., Chinnakotla, M. K., & Shrivastava, M. (2015, May). " Answer ka type kya he?" Learning to Classify Questions in Code-Mixed Language.

In *Proceedings of the 24th International Conference on World Wide Web*, pp. 853-858. https://doi.org/10.1145/2740908.2743006.

Ravikiran, M., & Annamalai, S. (2021). DOSA: Dravidian code-mixed offensive span identification dataset. In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pp. 10-17.

Roy, R. S., Katare, R., Ganguly, N., Laxman, S., & Choudhury, M. (2015). Discovering and understanding word-level user intent in web search queries. *Journal of Web Semantics*, *30*, 22-38. https://doi.org/10.1016/j.websem.2014.07.010.

Sarma, N., Singh, S. R., & Goswami, D. (2018). Word level language identification in Assamese-Bengali-Hindi-English code-mixed social media text. IEEE, In *2018 International Conference on Asian Language Processing (IALP)*, pp. 261-266. https://doi.org/10.1109/IALP.2018.8629104.

Sasidhar, T. T., Premjith, B., & Soman, K. P. (2020). Emotion detection in hinglish (hindi+ english) code-mixed social media text. *Procedia Computer Science*, *171*, 1346-1352. https://doi.org/10.1016/j.procs.2020.04.144.

Shanmugalingam, K., Sumathipala, S., & Premachandra, C. (2018). Word level language identification of code mixing text in social media using NLP. *IEEE*, In *2018 3rd International Conference on Information Technology Research (ICITR)*, pp. 1-5. https://doi.org/10.1109/ICITR.2018.8736127.

Sharma, V. K., & Mittal, N. (2018). Cross-lingual information retrieval: A dictionary-based query translation approach. In *Advances in Computer and Computational Sciences: Proceedings of ICCCCS 2016, Volume 2*, pp. 611-618. Springer Singapore. https://doi.org/10.1007/978-981-10-3773-3_59.

Shashirekha, H. L., Balouchzahi, F., Anusha, M. D., & Sidorov, G. (2022). CoLI-machine learning approaches for code-mixed language identification at the word level in Kannada-English texts. *arXiv preprint arXiv: 2211.09847*. https://doi.org/10.12700/APH.19.10.2022.10.8.

Shekhar, S., & Sharma, D. K. (2020). Computational intelligence for temporal expression retrieval in code-mixed text. IEEE, In *2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC)*, pp. 386-390. https://doi.org/10.1109/PARC49193.2020.236634.

Shekhar, S., Sharma, D. K., & Beg, M. S. (2018). Hindi roman linguistic framework for retrieving transliteration variants using bootstrapping. *Procedia Computer Science*, *125*, 59-67. https://doi.org/10.1016/j.procs.2017.12.010.

Shekhar, S., Sharma, D. K., & Beg, M. S. (2020). Language identification framework in code-mixed social media text based on quantum LSTM—the word belongs to which language? *Modern Physics Letters B*, *34*(06), 2050086. https://doi.org/10.1142/S0217984920500864.

Sristy, N. B., Krishna, N. S., Krishna, B. S., & Ravi, V. (2017). Language identification in mixed script. In *Proceedings of the 9th Annual Meeting of the Forum for Information Retrieval Evaluation*, pp. 14-20. https://doi.org/10.1145/3158354.3158357.

Thara, S., & Poornachandran, P. (2018). Code-mixing: A brief survey. IEEE, In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2382-2388 https://doi.org/10.1109/ICACCI.2018.8554413.

Velankar, A., Patil, H., & Joshi, R. (2022). A review of challenges in machine learning based automated hate speech detection. *arXiv preprint arXiv: 2209.05294*.

How to cite this Article:

Anu Chaudhary, Rahul Pradhan and Shashi Shekhar (2024). A Novel Framework for Multilingual Script Detection and Pattern Analysis in Mixed Script Queries. *International Journal of Experimental Research and Review*, *43*, 214-228.

DOI : https://doi.org/10.52756/ijerr.2024.v43spl.016