Jestr

Journal of Engineering Science and Technology Review 17 (1) (2024) 63 - 70

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Language Identification and Transliteration approaches for Code-Mixed Text

Madhuri Kumbhar* and Kalpana Thakre

Marathwada Mitra Mandal's College of Engineering, Pune, India

Received 24October 2023; Accepted 11January 2024

Abstract

People have become part of the digital era with the advent of the Web. They actively create, share, a variety of content on the web. Unlike earlier days, people widely use different social platforms to talk about their interests, hobbies, reviews on movies, and purchased items in natural language. Processing such natural languages with mixed language tasks is challenging. A sizable proportion communicates in regional language but using code-mixed and script like Roman and Devnagari for English and Marathi language. These texts are generally informal, causal, short length, non-standard spelling alteration etc are prime challenges in language processing. Language identification in mixed text is challenging, since the Romanized string of several languages is comparable. Mixed text is essential to transform into native script for further processing like Information Retrieval, machine translation, Question Answering etc. Due to the lack of orthography of Latin script in Marathi, language modelling, and identification of mixed text is a challenging issue. Many NLP (Natural Language Processing) applications ranging from machine translation and information retrieval uses Machine transliteration as input mechanism for non-roman script. In this paper, different techniques and various approaches presented by the researchers for code-mixed language, Indian regional languages processing are discussed. The tasks like language identification, transliteration, Named Entity Recognition are reviewed with respect to Statistical, Rule and Neural based approaches.

Keywords: Language Identification, Transliteration, Natural Language Processing, Indian regional languages, Code-Mixed Text.

1. Introduction

People use colloquial language using mixed language and script to express them. Recently, the majority of research on social media texts was conducted in English. But today a remarkable percentage of articles, posts, and news are in languages other than English. Many times people prefer to use their local language to express their views, and comments along with the English language. This text might be a combination of local language script along with Roman Script. Data generated on digital media needs linguistic analysis and tools for understanding data especially when text is Code-Mixed. These texts are generally informal, casual, and short in length, non-standard spelling differences, generating major challenges in processing [1]. Text-to-speech conversion, Chatbot, Sentiment Analysis, Language Recognition, Spelling checking, e-medical records summarization, and many other applications are being developed to handle natural languages for real-time requirements. This paper addresses code-mixed text written in Marathi and English. Marathi is an official language of India, spoken largely by people living in the state of Maharashtra. When the script of one language is applied for writing another one, the result is a code-mixed language. The main task with such mixed text is to determine the language used for every word for further process.

Consider a following code-mixed sentence:

"kalcha movie khup सुंदर होता, tu pan bagh nakki"

The above sentence is an example of Marathi-English code-mixed text. English words are used together with Marathi words. 'movie' is an English word, kalcha, khup, tu, pan, bagh, and nakki are Marathi words written in Roman script, and 'सुंदर' and 'होता' are Marathi words in Devnagari script. Code-mixed data should be processed for further analysis. It needs to be transliterated into a single script and understanding code at the semantic level is very important for information retrieval, knowledge acquisition, semantic interpretation, etc. Each word should be correctly transliterated with its correct sense. This paper discusses different issues of language identification and transliteration, back transliteration of Indian Mixed-code text.

For Example:

Table 1. Mixed-code Indian language

Mixed Code Sentence	Language	Script
Mazya kadil tulshichi pane ashi pivalsar hotahet ani tyavar black dots yetahet. Kashyamule asave ani upay kay karava?	Marathi	Roman
mera naam Raju hai. mai ek computer engineer hue.	Hindi	Roman

In Table 1, mixed-code sentences consist of some words are English words, many are Marathi words written in Roman script, but the language of these sentences is Indian. Here, only two languages are considered for example

*E-mail address: kumbharmadhuri@gmail.com
ISSN: 1791-2377
doi:10.25103/jestr.171.09

purposes. Such mixed-code text needs language processing to analyse it further.

Language processing techniques are broadly explored for the English language. However, moderate work has been reported for Indian languages, as they are rich in morphology and complexity in structure. This paper reviews various techniques and approaches for language identification and transliteration.

The subsequent portions of the paper have been organized: Section 2 discusses general natural language processing on Indian regional languages and mixed languages and scripts. Section 3 discusses various approaches used for language identification for mixed language and code. Section 4 reviews various transliteration processes introduced for Indian languages. Section 5 presents an overall review of language identification and transliteration based on different parameters.

In recent years, the internet has not remained as monolingual; contents in regional languages are growing rapidly [2]. Amarappa [3] has mentioned that a lot of research has been conducted to make it easier for users to interact with computers in region-specific natural languages. Google Translate offers transliteration in Indian Local Languages such as Tamil, Bengali, Punjabi, Marathi, Kannada, Hindi, Telugu, Malayalam and Gujarati [4]. Machine Translation, grammatical tagging, Sentiment Analysis, and Named Entity Recognition are the main activities focused on Indian Regional languages. Machine translation is the technique of utilizing artificial intelligence to automatically convert text from a single language to the other without the assistance of a human. A tag is assigned to each word in a sentence that specifies its relevant part of speech in POS Tagging. The proper names in documents are identified in Named Entity Recognition and then names are classified into sets of predefined categories as per interest. Native languages can be found all over the world, each with a distinct alphabet, symbols, grammar, and signs. There is an ancient and morphologically distinct range of regional languages in India. Using common ASCII codes, data expressed in English is easier for computer processing than data represented using different regional natural languages.

Preprocessing techniques are used in natural language processing after tokenization and transliteration process to increase efficiency of NLP applications like Stemming, Stop-word removal, Lemmatization, POS Tagging, Unicode Normalization etc. The model is trained after preprocessing using rule based language processing or statistical approaches. To extract features from preprocessed data, the machine learning approach employs feature extraction. Furthermore, neural-based methods yield better outcomes, where artificial neurons are used to process the information that has been given. Harish and Raghavan [5] studied Indian regional languages processing and concluded that neural-based method yields better results in a variety of complicated circumstances such faster and better learning, training large datasets, presence of features like learning ability, uniformity, generalization ability and computation power for language processing tasks.

2. Generic Natural Language Processing

2.1 Indian Languages

A new research study by Google and KPMG in 2021 found that, Hindi will overtake English as India's most popular web language, while Marathi, Bengali, Telugu, and Tamil will account for 30% of the country's overall local language user population. According to one survey, in India, native languages are used by 234,000,000 internet users, in comparison to 175 million English end-users. This figure is likely to rise to 534,000,000 in the following three years. 90% of newly registered web users in the country opt to communicate on social media in their original language [2].

2.2 Code-Mixed Language

Code-mixed communication is a method of talking with people in short bursts of text as well as efficiently conveying one's own views. Because switching within keyboard interfaces is inconvenient, such code-mixing is typically written in a similar Romanized script. Romanized Text Processing: Language identification in Romanization is challenging, as many languages share similarities in their Romanized string. Romanized text is essential to transform into native script (Devnagari) for further processing like Information Retrieval, machine translation, Question Answering etc. Due to the lack of orthography of Latin script in Marathi, language modelling, and identification of Romanized text is a challenging issue [6]. People express their opinion or interest on Social media platforms. Previously, the only language used for social media was English. However, the blending of multiple languages together is becoming more common in Code-mixed text.

2.3 Approaches for Language Processing

2.3a Rule based NLP

In a rule based method, the language processing activities are completed on the basis of morphological analysis, lexical rules and linguistic knowledge pre-processing. The decisions made by the lexical rules handle the language processing tasks. Language processing tasks must be completed efficiently by taking into account the linguistic laws of each language. Rules are also frequently employed in text preparation, which is required for ML-based NLP. For example, rules can successfully execute tokenization where text is divided into words and part-of-speech tagging where nouns, verbs etc. are labelled.

2.3b Statistical or Machine Learning based NLP

The preprocessed data are analysed with statistical metrics in the statistical-based strategy to reach the intended outcome in language processing. Machine learning (or statistical) methods to NLP employ AI algorithms to address problems without explicit programming. Rather than handling human-written patterns, ML models discover them on their own through analysing texts. This approach looks for statistical relations like distance metric, probability metric, etc. in preprocessed data.

2.3c Deep learning based NLP

There has already been an enormous amount of work done in recent years with neural-based language processing solutions. Some language processing tasks with neural networks produced improved results; nevertheless, due to resource scarcity, a little study into regional languages produced average results. The RNN (Recurrent Neural Network) and LSTM (Long Short-Term Memory) models can also be used for developing learning models. NMT (Neural machine translation) refers to a neural-based machine translation process. The encoder-decoder technique is used by the majority of the proposed NMT. Bahdanau et al.[7] had developed a system where the variable input

sentence is converted into a fixed length vector by an encoder, the decoder then translates this into the target language sentence.

3. Language Identification

The language identification at word level with Marathi-English mixed mode scenarios is reviewed in this study. The language and script detection at word level in code-mixed text, where few words are written in roman and few in Devnagari is the first important step in language processing. The language of text is determined by examining short text fragments from a set of languages. The major stage in monolingual text processing would be POS labelling of the text whereas in code-mixed text the primary task is language identification. To represent each word, word embedding features can be used.

Further down, existing language identification approaches are surveyed with different parameters.

Sarkar [2] presented the Unicode Detection Analysis technique for tweet classification to respective scripts of Hindi, Bengali, and English languages. Here, the Unicode values of the tweets are compared with the script's own Unicode range for script detection. Irrespective of the script used in tweets, this system automatically identifies and classifies native tweets. Here, text which is detected as English by the unicode detection system is further processed with SVM and naive bayes classifiers. This classification identifies Hindi and Bengali words written in Roman script. SVM classifier has used 3- gram of character to classify English words. Along with Script Identification, this approach consists of language analysis and Clustered mining. To classify tweets to their respective scripts, they used Unicode detection analysis.

Zhang et al. [8] described CMX, a fast and compact model for fine-grained language detection. It outperforms comparable models on code-mixed and monolingual texts, which are composed of many datasets ranging text in a number of languages and obtained from various sources. They constructed and evaluated a code mixed corpus for 25k code mixed sentences. This corpus comprises user-generated posts in English mixed with Indonesian, Spanish and Hindi. CMX contains three features: character n-grams, scripts, and lexicons. In this system, on monolingual inputs, the lexical features of CMX yield a 2.0% absolute improvement in accuracy.

Ding [9] proposed an accurate language detection method for social media short text utilizing a Vectorization-based approach and an upgraded Cascade Forest method. A line of text is represented by a fixed-length feature vector, and it is then assigned a language using a supervised classifier. This approach consists of two parts: 1. Representation of features and 2. Identification of Language. Weight frequency of each sentence is calculated using character n-gram instead of word n-gram. This experiment has used TATOEBA (https://tatoeba.org/eng/downloads) and twitter dataset for language identification. Compared to cutting-edge machine learning algorithms this approach yields higher precision, accuracy, and recall rates. They looked at two ways of representing text features: N-gram statistical features and distributed sentence representations.

Kazi and Mehta [1] presented language identification at sentence level for code mixed script in Gujarati where Gujarati is spelled in both Romanized and non-Romanized forms. Here, a dataset containing Hindi, English mixed with Gujarati is being used for study. Distinct classifiers like Naive Bayes, k-nearest Neighbor, SVMx, Random Forest, Decision Tree, and Logistic Regression are employed in the creation of language identification systems. These classifiers performed far better than statistical approaches. The system employed manually built a multilingual corpus of code-mixed Hindi, Gujarati and English texts.

Roark et al. [6] described the Dakshina dataset included Latin and native script text in support of 12 south Asian languages. Here database contains native script Wikipedia content, a lexicon in Roman, and complete sentence parallel data for each language that includes a native script and a Latin alphabet. This dataset consists of baseline results during different tasks like language modeling of Romanized text and native script and single word and full sentence transliteration. Various modeling methods suggested that different tasks like language modeling, single word transliteration, and complete sentence transliteration served in this system are challenging and needed further research.

Bhargava [10] has extracted opinions, and sentiments starting with mixed script sentences. Initially language of each word was identified using Machine learning techniques subsequently tagged sentences underwent sentiment analysis with language-specific SentiWordNet. The suggested approach additionally divides analysis tasks into three steps: language labelling at a word level, Indic languages back-transliteration, and sentiment analysis of queries. In this step Hindi/ Telugu/ Telugu/ English/ Tamil/ Bengali language is identified from the mixed sentences, Romanized script is transliterated into its Indic script, and the statement's sentiment is judged word-wise. Also developed an approach for code mixed sentences' sentiments mining considering English, Tamil, Hindi, Telugu, and Bengali in two phases. The first phase is the identification of language and the second is sentiment analysis applying various tools of machine learning. Here, texts are converted to their corresponding regional languages using Google transliteration. This is a fairly divergent technique because each regional language has its own parser and language specialist. After applying a naive Bayes classifier, n-grams are employed to improve accuracy at the word level.

A hybrid technique to word-level language detection for Romanized Bangla words mixed with English words was proposed by Banerjee [11]. Here, for training and testing, both manual and automated procedures are employed. The word-level language identification is developed by using Conditional Random Field (CRF). The task of language identification has been broken into two parts: word-level language categorization and transliteration of detected Indian language words into native script. They discovered that employing simple post-processing heuristics improves overall efficiency of the word-level language approach.

Barman [12] analyzed distinct approaches for language identification inclusive of dictionary based, word level classification with SVM, sequence labelling with (CRF) conditional random field. It was concluded that the CRF model surpasses all other approaches. Also expressed an automatic word-level language identification approach for code mixed Hindi, Bengali, English languages. Word-level classification is tested on manually formed Bengali-Hindi-English Facebook comments dataset.

Sristy [13] put forward two level proposals for word level language identification using CRF, Naïve Bayes classifiers and logistic regression. The first stage recognizes mixed language by employing sentence character n-grams. This mixing combination class is used in the second stage

for word level language identification. Here English and non-English words are distinguished by binary classifier. Second stage of Conditional Random Fields applied to enhance word level language identification performance. This system uses the FIRE 2015 dataset that involves languages like Kannada, Hindi, Tamil, Gujarati, Telugu, and Malayalam. All of these languages have been mixed in with English. Due to a lack of Gujarati datasets, the system concluded that hardly any of the algorithms could detect the Gujarati language.

Eskander [14] has described the system of Arabic language word identification. Here Arabizi input processing is handled by code switching. This approach determines whether each word belongs to one of four categories (Arabic, name, punctuation, and sound) or foreign word. This system made use of a feature set that included various n-grams combinations, the word, word length, and certain probabilistic factors. They achieved system performance of 83.8% on unseen test data. Shekhar and Sharma [15] proposed cBoW and Skip gram model based deep learning framework for language identification of code-mixed data with Hindi-English language utilized across a social media network. Here, each word was represented using popular word embedding techniques. Multichannel neural networks with CNN and BLSTM are used to determine word level language in English and Hindi code-mixed data. According to experimental results, accuracy of the word-based model succeeds over the character-based embedding model. Additionally, it showed that 3-gram-based features beat 1-gram and 5-gram context feature methods in the word-based embedding model. In the character-based model 5-gram features outperform 1-gram and 3-gram context feature methods.

Das and Gambäck [16] identified languages including English-Hindi and English-Bengali in the Indian context. This system applied n-grams having weights, minimal edit distance-based weight features, dictionary-based and word context features with support vector machine using a linear kernel. This study used corpus containing mixed English Bengali and English Hindi to find word level language boundaries. This language identification system mostly employs common approaches such as character n-grams, dictionaries, and SVM-classifiers. Patel [17] studied English, Gujarati linguistic resources. This method demonstrated data normalization, language identification, and translation of the transliterated text into the native language. They started by creating a dictionary of Gujarati words with the proper variations. They used the look-up method to deal with word variations and detect Gujarati language when combined with another language.

Shanmugalingam [18] applied word level automatic language identification with machine learning algorithms. Tamil and English code-mixed corpora is studied where Tamil characters were transcribed in Romanized character and combined in with English. Here, training and testing is implemented using different machine learning classifiers such as Logistic Regression, Naive Bayes, Decision Trees, Random Forest and Support Vector Machines. The classifier is trained using features like Tamil Unicode letters in Roman, dictionaries, double consonants, and term frequency. Here, the SVM classifier achieved the best accuracy of 89.46%. Veena and Anand Kumar [19] identified language for Hindi - English code mixed text using Support Vector Machine. This system used data in three social media platforms like Twitter, Facebook and WhatsApp. The mixed text was classified into Named Entity, Hindi, English, mixed Hindi-English, Acronym, Universal, and undefined tags. Here, each word is represented by word embedding features like character-based context and word-based embedding features.

Gambäck and Das [20] offered an enhanced version for measuring the code mixing complexity level in a multilingual code-mixed corpus. Here, the proposed system uses Code-Mixing index to determine the difficulty of texts written in multiple languages. They also concentrated on an issue that is especially prevalent with high social media writing in geographical areas with high percentages of bi-lingual and multilingual residents, like the Indian subcontinent. Gupta and Raghuwansh [21] proposed an approach that detects the source of the word in the sequence from the language's viewpoint depending on the particular words that precede it orderly. As compared with character embedding, this model provides superior accuracy for word embedding. Here, two systems designed around word-based embedding features with character-based context features. A technique considering character-based systems is similar with word based system; aside from the vectors are character vectors.

Lakshmi [22] concentrated on code mixed data considering word level language identification. A dataset consists of mixed sentences in English and Kannada language from social platforms. Different supervised classifiers are used here with integrating a dictionary module to operate word level identification. They have considered code mixed sentences like inter and intra sentential code and Word-level mixed code. They included corpora that contain code mixed sentences from social platforms with more than 6000 English words, 6250 Kannada words and 500 mixed words. Classification algorithms perform language identification with inter and intra-sentential code mixing. Dictionary lookup module is used for Word-level code mixed sentences.

Sharma et al. [23] proposed a new semi-supervised method based on deep learning techniques primarily neural networks for Language Identification. This approach has applied specifically in the Hindi Language composed in Roman Script with Generative Adversarial Networks. This model works with consistent accuracy when the number of words in the dataset per example exceeds ten and with adequate accuracy when the number of words is less than ten. In this work, a complicated dataset containing a combination of cursive scripts (Hindi, Bengali, Saraiki) and non-cursive scripts (English and Roman Urdu) is used. According to observations made here, accuracy drops as the total count of languages with the cursive mixed script database grows.

4. Transliteration

The primary language of India is Hindi, and it is widely utilized by about 500 million people. After English, Chinese, Spanish, Hindi is the world's fourth commonly used spoken language. Marathi is extremely common dialect in India, particularly Maharashtra state. Hindi and Marathi are Sanskrit-derived languages that employ "Devnagari" alphabet for writing. Transliterating vocabulary terms that occurs in user input within languages with different alphabets and sound inventories is tough. The technique of converting a word in one language into another while keeping its phonetic qualities is known as transliteration [24]. Hindi to English and Marathi to English Named Entity

transliteration is Many considerations, including variations in writing script, alphabet count, and capitalization of leading letters, phonetic features, length of character, number of correct transliterations, and the presence of the parallel corpus, make it challenging [25].

Transliteration changes one alphabet or language into the corresponding, similar-sounding characters of another alphabet. Transliteration tasks become difficult in presence of out of vocabulary words and noisy words which are present in the corpus of documents. The transliteration of word "राष्ट्रपती" with various combinations may possible like "rashtrapati", "rashtrapathi", "raashtrapathy", "raashtrpati". Among all accessible transliteration systems, Google Input Tools provides the finest Marathi transliteration. Input tool of Google transliterate a single Romanized Marathi word at a time to Devnagari script as follow:

1. Enable Input Tools and type the word by involving the sounds, production, or transcription of speech in Latin characters.
2. Here displays a list of word candidates mapping to the phonetic spelling. Here, the word must be preceded by a 'Space', /'Enter'/'Tab' key.
3. Choose a word from the list.

But transliteration of one or more sentences with mixed language and script with correct sense is erroneous with this tool. Existing transliteration generation approaches are briefly explained in the following section.

1. Grapheme based approaches

Transliteration transfers a character sequence from one language to another while neglecting phoneme-level processes. The grapheme-based technique transliterates the text by mapping the language.

2. Phoneme based approaches

The phoneme-based technique [26] matches the source language text to its phonemes. The phonemes are later mapped to the phonemes of the destination language. Finally, the destination language's phonemes are mapped to alphabets of the destination language.

3. Corpus based approach

This approach not merely allowed for transliteration without diacritic marks, even so it was further useful in selecting the correct word regarding numerous feasible transliterations employing probabilistic analysis.

The rightness of grapheme-based transliteration is fairly modest because it overlooks phonetic and transcription issues in any language's written text.

Here, existing transliteration approaches are reviewed with different parameters.

Sarkar and Sinhababu [2] built a transliteration API for tweets in JavaScript with Online Google Input Tools. They concentrated on Twitter data as tweets are brief. Here, sentence-level classifications are performed, and every sentence regardless of language or script is translated to English for analysis. Here, translation is performed when the text is in Bengali or Hindi script and when Bengali or Hindi text is in Roman script. The naïve Bayes classifier and SVM were applied for language processing in the Bengali, English, Hindi languages primarily with Roman script. They have designed an ensemble algorithm by considering the positive features of both classifier's positive features. The accuracy of this designed system is high and learning rate is also steep

Banerjee et. al. [11] implemented a model at syllable-level and chunk-level for transliterating Bangla words drafted in Roman script to Bangla script. They built a transliteration system by segmenting English and Bangla words into pieces of subsequent characters and training the algorithm with this segmented data. For this, two approaches are used. In first, words break into pieces of consecutive 2/3 characters and in second words are broken into transliteration units following the heuristic. On the basis of transliteration data segmentation, they have developed two types of the transliteration systems, namely at the chunk-level and syllable-level.

Abbas and Asif [27] proposed a transliteration system for Punjabi Gurmukhi script into Latin script with accuracy approximately 96.82%. This system used phonetic rectification for Punjabi script and then applied character-to-character mapping. Also to insert schwa, the system formulated a generic finite state transducer rather than the fixed syllabic pattern.

Dhore and Dhore [28], applied a Linguistic and Metrical approach for Indian-origin named entities' transliteration that uses a hybrid approach that combines metrical and linguistic rules. The phoneme concept of machine transliteration is used in the rule-based transliteration approach. Such a model made use of the phonetic mapping between the Marathi with Devnagari script (source language) and English with Roman script (target language). After phonetic mapping, schwa is removed on the basis of syllable stress. As compared to existing grapheme-based techniques, this approach provides superior absolute performance. Tyson [29] implemented schwa position detection and elimination by stress analysis based rules. They predicted schwa deletion for two-three syllable words using syllable structure and stress assignment. This achieved high accuracy of 95% on elimination of schwa from smaller corpus of Hindi words in text to speech synthesis. This method employed prosodic information to determine whether schwa should be deleted.

Dhore and Dixit [30] suggested a transliteration tool based on stress analysis and phonology. They proposed a phonetic model for transliterating names of Indian origin into English utilizing the full consonant technique. This study shows that a thorough understanding of the creation of words in Devnagari script-based languages beats statistical techniques. For schwa elimination, this model employs hybrid techniques that integrate rule-based and metric-based stress analysis. Deep and Goyal [31] showed Punjabi to English hybrid transliteration from Gurumukhi script to Roman script for person names. They have developed 41 rules for transliteration and these rules are applied on each tokenized word. If neither of the rules applies to the word, direct mapping is used. They achieved overall accuracy of 95.23%.

Ngoc and Fatiha [32] explained transliteration for French to Vietnamese using Grapheme-to-Phoneme alignments. The transliteration system consists of steps like preprocessing, input sequences are modified based on alignment representation, and RNN-based machine transliteration. This system uses just a mini bilingual pronunciation dictionary. Data in this dictionary are preprocessed with normalization in lowercasing, syllable separation by removing hyphens, and character level syllable segmentation. Here, input sequences and extracted alignment output derived from bilingual pronunciation dictionaries are altered based on the alignment results. This transliteration system was experimented on French-Vietnamese low-resource language

and appropriate only for monolingual pronunciation dictionaries.

Prabhakar and Pal [33] described the customized phonetic matching transliteration approach for Hindi language written in Roman script. Character-wise mapping of Editex and Soundex, as well as Levenstein edit distance is used to design this approach. This algorithm measures dissimilarity between two terms. A specific set of rules for phonetic encoding of transliterated Hindi words is also designed. These rules are devised based on observations of Hindi words representation, consonants and vowels sound, and basic knowledge of Soundex, Editex, and Phonix. Designed encodings aided in identifying phonetically matching Hindi lyrics, which can be used to improve retrieval accuracy.

Ngo and Nguyen [34] explained the Phonology-augmented transliteration statistical model that integrates phonological knowledge of the target language distinctly with a statistical model. The pseudo-syllables enforce phonological constraints of syllable structure in target language. This system performs transliteration from Vietnamese to Cantonese language with the following steps: In the first step, source word graphemes are arranged into pseudo-syllables. Source graphemes are explicitly assigned to the sub-syllabic parts in every pseudo syllable. In the second step, to trace the graphemes to each pseudo-syllable to most repeated phonemes, a language model is applied. In the last step, according to the target language's phonemes in every syllable, one tone is allocated to each syllable. Ravishankar [35] constructed back transliteration of roman text for Indian language using a finite-state based system. Finite-state transducer is used here to transliterate Romanagari Marathi to formal Devnagari. Here, the author had created their own corpus for evaluation which was a combination of three mini corpora.

Chinnakotla and Damani [36] proposed a system for transliterating from Hindi to English and vice versa. They demonstrate how to develop a workable transliteration procedure for languages with limited resources that lack the sizable parallel corpora necessary for training statistical techniques. They noticed that 3-grams work best for determining the Hindi word's origin. They used n-gram character sequence modeling to predict the correct alphabet sequence. This system uses Character Sequence Modeling (CSM) language model for word origin identification for source language. Character sequence modeling used in their system eliminates inconsistencies with character-based n-gram models, but transliteration could result in invalid order of alphabets in target language.

A rule-based back-transliteration method for Sanskrit script that converts ASCII-encoded English to Devnagari according to Harvard-Kyoto (HK) standard was proposed by Nair and Sadasivan [37]. This work also discusses the numerous standard methods for transcribing Devnagari into Roman. They advocated stepping up their efforts to create comparable tools for other Indian languages that use Devnagari script, such as Marathi and Hindi. For back-transliteration, they used a rule-based and grapheme-based alphabet alignment model. Jong-Hoon [38] provided a transliteration model that particularly dynamic utilizes graphemes as well as phonemes, with a focus on their interrelation. In Phoneme-based technique, the source language text is converted to its phonemes, which are then mapped to target language's phonemes. Target language's phonemes are finally translated into its alphabets. They produced better results than other models, with improvements ranging from 15- 41% in English-Korean transliteration and 16-44% in English-Japanese transliteration.

5. Discussion

In multilingual communities, code-mixing is a natural happening. It facilitates communication and allows for a broader range of expression, making it a common style of expression in social media interactions. These texts are generally informal, casual, short in length, and with non-standard differing spellings. These are major challenges in mixed-code text processing. To process code-mixed Marathi-English text, each word in a sentence needs to be identified and labelled in the correct language. Many approaches studied for mixed-code language identification used machine learning approaches with datasets of code-mixed text. Due to unavailability of Indian language mixed-code corpus, researchers have to create and annotate corpus manually. Language identification will produce correct results with machine learning algorithms if models are trained accurately with mixed- code text. Some approaches perform identification of language at sentence level and some at word level using n-gram features. Very less work has been done for Marathi-English code-mixed language processing. So in the future, attention would like to investigate better techniques for word-level identification tasks for Marathi-English code-mixed data.

Transliteration is one vital subtask in the Machine translation process for mixed text documents. There are approaches like phonetic mapping that are applied for transliteration. It is observed that to analyze and process English-Marathi code-mixed data for information retrieval, customer review identification, knowledge acquisition, semantic interpretation, etc, needs to be transliterated into a single script, and understanding code at the semantic level is critical. The rule based approach for transliteration will be more effective for Indian languages. It has been observed that less training data is available for transliteration. So, statistical models face challenges while phonological rules learning.

6. Conclusion

The most recent language identification and transliteration techniques for mixed language and script are thoroughly discussed here. Code-mixed text, specifically Marathi, Hindi, English, and Roman and Devnagari script combinations, is commonly used by the Marathi community. This study focuses on a detailed discussion of numerous research projects for language identification and transliteration in Indian languages. In view of code mixed Marathi-English text, there needs to be more study for processing and analysis of language. In the future, we will aim to develop a better transliteration method for English-Marathi code-mixed text. These details will be useful for future epochs of the regional language research community.

References

[1] M. Kazi, H. Mehta, and S. Bharti, "Sentence level language identification in Gujarati-Hindi code-mixed scripts," *IEEE Int. Sympos. Sustain. Ener., Sign. Proces. Cyb. Secur. (iSSSC)*, pp.1-6, Dec. 2020, doi: 10.1109/iSSSC50941.2020.9358837

[2] B. Sarkar, N. Sinhababu, M. Roy, P. K. D. Pramanik, and P. Choudhury, "Mining multilingual and multiscript Twitter data: unleashing the language and script barrier," *Int. J. Bus. Intell. Data Min.*, vol. 16, no. 1, pp. 107-127, Nov. 2019, doi: https://doi.org/10.1504/IJBIDM.2020.103847.

[3] S. Amarappa and S. Sathyanaryana, "Kannada Named Entity Recognition and Classification (NERC) Based on Multinomial Naïve Bayes (MNB) Classifier," *Int. J. Nat. Lang. Comp.*, vol. 4, no. 4, pp. 39–52, Aug. 2015, doi: 10.5121/ijnlc.2015.4404.

[4] M. Madankar, M. B. Chandak, and N. Chavhan, "Information Retrieval System and Machine Translation: A Review," *Proc. Comp. Sci.*, vol. 78, pp. 845–850, 2016, doi: 10.1016/j.procs.2016.02.071.

[5] B. S. Harish and R. K. Rangan, "A comprehensive survey on Indian regional language processing," *SN Appl. Sci.*, vol. 2, no. 7, p. 1204, Jul. 2020, doi: 10.1007/s42452-020-2983-x.

[6] B. Roark *et al*., "Processing South Asian Languages Written in the Latin Script: the Dakshina Dataset," *Proceedings of the Twelfth Language Resources and Evaluation Conference*, Marseille, France, pp. 2413–2423, July 2020,doi: https://doi.org/10.48550/arXiv.2007.01176.

[7] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," 2014, doi: 10.48550/ARXIV.1409.0473.

[8] Y. Zhang, J. Riesa, D. Gillick, A. Bakalov, J. Baldridge, and D. Weiss, "A Fast, Compact, Accurate Model for Language Identification of Codemixed Text," in *Proc. 2018 Conf. Empir. Meth. Nat. Lang. Process.*, Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 328–337. doi: 10.18653/v1/D18-1030.

[9] B. Ding, H. Luo, Z. Wu, and S. Zhang, "A Vectorization Approach to Language Identification of Social Media Short Texts," in *Proc. 2020 12th Int. Conf. Mach. Learn. Comp.*, Shenzhen China: ACM, Feb. 2020, pp. 462–467. doi: 10.1145/3383972.3384020.

[10] R. Bhargava, Y. Sharma, and S. Sharma, "Sentiment analysis for mixed script Indic sentences," in *2016 Int. Conf. Adv. Comp., Commun. Informat. (ICACCI)*, Jaipur, India: IEEE, Sep. 2016, pp. 524–529. doi: 10.1109/ICACCI.2016.7732099.

[11] S. Banerjee, A. Kuila, A. Roy, S. K. Naskar, P. Rosso, and S. Bandyopadhyay, "A Hybrid Approach for Transliterated Word-Level Language Identification: CRF with Post-Processing Heuristics," in *Proceedings of the Forum for Information Retrieval Evaluation on - FIRE '14*, Bangalore, India: ACM Press, 2015, pp. 54–59. doi: 10.1145/2824864.2824876.

[12] U. Barman, A. Das, J. Wagner, and J. Foster, "Code Mixing: A Challenge for Language Identification in the Language of Social Media," in *Proceed. First Worksh. Computat. Approach. Code-Switchin*, Doha. Qatar. Oct. 2014, pp. 13 – 13, doi: 10.13140/2.1.3385.6967.

[13] N. B. Sristy, N. S. Krishna, B. S. Krishna, and V. Ravi, "Language Identification in Mixed Script," in *Proc. 9th Ann. Meet. Forum Inform. Retr. Eval.*, Bangalore India: ACM, Dec. 2017, pp. 14–20. doi: 10.1145/3158354.3158357.

[14] R. Eskander, M. Al-Badrashiny, N. Habash, and O. Rambow, "Foreign Words and the Automatic Processing of Arabic Social Media Text Written in Roman Script," in *Proceed. First Worksh. Computat. Approach. Code-Switchin.*, M. Diab, J. Hirschberg, P. Fung, and T. Solorio, Eds., Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1–12. doi: 10.3115/v1/W14-3901.

[15] S. Shekhar, D. K. Sharma, and M. M. Sufyan Beg, "An Effective Bi-LSTM Word Embedding System for Analysis and Identification of Language in Code-Mixed social Media Text in English and Roman Hindi," *CyS*, vol. 24, no. 4, Dec. 2020, doi: 10.13053/cys-24-4-3151.

[16] A. Das and B. Gambäck, "Identifying Languages at the Word Level in Code-Mixed Indian Social Media Text*," in Proc of the 11th International Conference on Natural Language Processing,* Goa, India. Citeseer, pp. 378–387, Dec. 2014.

[17] D. Patel and R. Parikh, "Language Identification and Translation of English and Gujarati code-mixed data," in *2020 Int. Conf. Emerg. Tren. Inform. Techn. Engin. (ic-ETITE)*, Vellore, India: IEEE, Feb. 2020, pp. 1–4. doi: 10.1109/ic-ETITE47903.2020.410

[18] K. Shanmugalingam, S. Sumathipala, and C. Premachandra, "Word level language identification of code mixing text in social media using NLP" in 3rd*Inter. Conf. Inform. Techn. Res. (ICITR) IEEE.* pp. 1-5, Dec. 2018, doi: 10.1109/ICITR.2018.8736127

[19] P. Veena and A. Kumar, "Character embedding for language identification in hindi-english code-mixed social media text" *Comput. Sis.*, vol. 22, pp. 65–74, Mar. 2018, doi: 10.13053/cys-22-1-2775

[20] B. Gambäck and A. Das, "On measuring the complexity of codemixing*" in 11th Intern. Conf. Nat. Lang. Process.*. Goa, pp.1-7, Dec. 2014.

[21] Y. Gupta, G. Raghuwansh, and A. Tripathi, "A New Methodology for Language Identification in Social Media Code-Mixed Text", *Adv. Intell. Sys. Comp.*, Singapore: Springer Nature Singapore, vol. 1141, pp.243-254, May. 2020, doi: https://doi.org/10.1007/978-981-15-3383-9_22

[22] B. S. Lakshmi, B. R. Shambhavi, "An Automatic Language Identification System for Code-Mixed English-Kannada Social Media Text," in *2nd Int. Conf. Comp. Sys. Infom. Techn. Sustain. Sol. (CSITSS),* pp. 1-5, Dec. 2017, doi: https://doi.org/10.1109/csitss.2017.8447784.

[23] D. K. Sharma, A. Singh, and A. Saroha, "Language identification for Hindi language transliterated text in Roman script using generative adversarial networks," in *Tow. Extens. Adapt. Meth. Comput.*, Singapore: Springer Singapore, pp. 267–279, Nov. 2018, doi: 10.1007/978-981-13-2348-5_20

[24] P. Nilesh, C. Manoj, N. Ajay, O.P. Damani, and P. Om, "Evaluation of Hindi to English, Marathi to English and English to Hindi," *Comp. Sci., Linguist.*, *CLIR at FIRE*, Jan. 2008.

[25] S. K. Saha, P. S. Ghosh, S. Sudeshna, and M. Pabitra, "Named entity recognition in Hindi using maximum entropy and transliteration." *Instituto Politécnico Nacional*, *Centro de Innovación y Desarrollo TecnológicoenCómputo, Polibits*," pp. 33–41, Oct. 2008.

[26] J. H. Oh, K.S. Choi, and H. Isahara, "A machine transliteration model based on correspondence between graphemes and phonemes," *ACM Trans. Asian Lang. Inf. Process.*, vol. 5, no. 3, pp. 185–208, Sept. 2006, doi: https://doi.org/10.1145/1194936.1194938.

[27] M. R. Abbas and K. H. Asif, "Punjabi to ISO 15919 and Roman Transliteration with Phonetic Rectification," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 19, no. 2, pp. 1–20, Mar. 2020, doi: 10.1145/3359991.

[28] M. Dhore and R. Dhore, "Marathi - English Named Entities Forward MachineTransliteration Using Linguistic and Metrical Approach," in *2017 Int. Conf. Comp., Commun., Contr. Automat. (ICCUBEA)*, Pune, India: IEEE, Aug. 2017, pp. 1–6. doi: 10.1109/ICCUBEA.2017.8463731.

[29] N. R. Tyson and I. Nagar, "Prosodic rules for schwa-deletion in hindi text-to-speech synthesis," *Int J Speech Technol*, vol. 12, no. 1, pp. 15–25, Mar. 2009, doi: 10.1007/s10772-009-9040-x.

[30] M. L. Dhore, S. K. Dixit, and R. M. Dhore, "Hindi and Marathi to English NE Transliteration Tool using Phonology and Stress Analysis," in *COLING 2012: Demonstr. Papers*, Mumbai, India, Dec. 2012, pp. 111–118.

[31] K. Deep and V. Goyal, "Hybrid Approach for Punjabi to English Transliteration System," *IJCA*, vol. 28, no. 1, pp. 1–6, Aug. 2011, doi: 10.5120/3356-4629.

[32] N. T. Le and F. Sadat, "Low-Resource Machine Transliteration Using Recurrent Neural Networks of Asian Languages," in *Proc. Seventh Named Ent. Worksh.*, N. Chen, R. E. Banchs, X. Duan, M. Zhang, and H. Li, Eds., Melbourne, Australia: Association for Computational Linguistics, Jul. 2018, pp. 95–100. doi: 10.18653/v1/W18-2414.

[33] D. K. Prabhakar, S. Pal, and C. Kumar, "Query Expansion for Transliterated Text Retrieval," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 20, no. 4, pp. 1–34, Jul. 2021, doi: 10.1145/3447649.

[34] G. H. Ngo, M. Nguyen, and N. F. Chen, "Phonology-Augmented Statistical Framework for Machine Transliteration Using Limited Linguistic Resources," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 27, no. 1, pp. 199–211, Jan. 2019, doi: 10.1109/TASLP.2018.2875269.

[35] V. Ravishankar, "Finite-State Back-Transliteration for Marathi," *The Prague Bullet. Mathemat. Linguist.*, vol. 108, no. 1, pp. 319–329, Jun. 2017, doi: 10.1515/pralin-2017-0030.

[36] M. K. Chinnakotla, O. P. Damani, and A. Satoskar, "Transliteration for Resource-Scarce Languages," *ACM Trans. Asian Lang. Inform. Proces.*, vol. 9, no. 4, pp. 1–30, Dec. 2010, doi: 10.1145/1838751.1838753.

[37] J. Nair and A. Sadasivan, "A Roman to Devanagari Back-Transliteration Algorithm based on Harvard-Kyoto Convention," in *2019 IEEE 5th Inter. Conf. Converg. Techn. (I2CT)*, Bombay, India: IEEE, Mar. 2019, pp. 1–6. doi: 10.1109/I2CT45611.2019.9033576.

[38] J.-H. Oh, K.-S. Choi, and H. Isahara, "A machine transliteration model based on correspondence between graphemes and phonemes," *ACM Trans. Asian Lang. Inform. Proces.*,, vol. 5, no. 3, pp. 185–208, Sep. 2006, doi: 10.1145/1194936.1194938.