Sentiment Analysis of Code-Mixed Text: A Comprehensive Review

W.A.S.C. Perera†, H.A. Caldera††
anneshehari@vau.ac.lk hac@ucsc.cmb.ac.lk
†Faculty of Technological Studies, University of Vavuniya, Vavuniya, Sri Lanka
††University of Colombo School of Computing, Colombo 07, Sri Lanka

Summary
Sentiment Analysis is the task of identifying and extracting the opinion expressed in a text to determine the writer's perception of an entity. Most of the research works regarding Sentiment Analysis are focused on monolingual languages such as English. Classifiers are failed within the context of the code-mixed text as the text is created by mixing more than one language, and it may consist of creative writing, spelling variations, grammatical errors, and different word orders. Hence Sentiment Analysis of code-mixed text is a challenging task. This paper presents a state-of-the-art in Sentiment Analysis of code-mixed text by discussing each concept in detail. The paper also discusses and summarizes the focused areas, datasets, techniques, limitations, and performances of the literature related to code-mixing.

Keywords:
Code-mixed, Natural Language Processing, Sentiment Analysis

1. Introduction

The internet is one of the biggest revolutions in communication technology which changed the way of communication and information sharing. Having the wide accessibility of social media platforms like Twitter, Facebook, and YouTube, people turn in to the web to search and share opinions. This has created a large amount of data for interpretation. Although a human can easily identify the feeling given by a text written in a known language, computers are not capable of interpreting natural languages. In that case, the sentiment analyzing technique can be used as it is capable of identifying the opinion hidden behind a text by using Natural Language Processing (NLP). Sentiment Analysis (SA), in other words, opinion mining is the process of identifying and extracting the attitudes expressed in a text to determine the writer's perception of an entity. SA is not only useful in social media monitoring, business, and politics but also in almost all fields since it gains in-depth insight into people's attitudes regarding trends, people, organizations, products, or services [1].

Since Internet users are from all over the world, they often bring their language, background, and culture to web communication. Although English is considered as the base language in web communications such as commenting and messaging, many people tend to use various other native languages as well. Even if they use the native language, most users do not use Unicode characters. Instead, they mix languages and use phonetic typing and lexical borrowing due to the simplicity. This concept is known as code-mixing and this is the latest trend in web communication. In simple words, code-mixing means the mixing of two or more languages or language varieties in speech. As a result of code-mixing, different variations of languages have emerged [2].

Although, it is simple and straightforward to extract the opinion of text written in English or Unicode characters, SA for code-mixed text is considered as one of the challenging tasks. The usual preprocessing techniques use for monolingual SA such as stemming, PoS, and morphological analysis are insufficient here since these types of code-mixed text usually do not write by following proper grammar and consist of creative writing [3]. As a result, the code-mixed text is different from user to user and does not have exact words with exact spellings as in monolingual languages. Some of the challenges of code-mixed text are lack of formal grammar, spelling variations, creative spelling, undetermined mixing rules, noise, nonstandard abbreviations, long processes, lack of linguistic resources available, etc. [3], [1]. Hence, SA of code-mixed text is an interesting, challenging and popular research field. However, a huge improvement can be observed in the field due to the help of advanced NLP tools and techniques.

This survey paper discusses and summarizes the concepts and literature in SA of code-mixed text including the levels of SA, approaches, challenges, performances, limitations, etc. The main contributions of the survey are: (a). Describe the generic process of SA, (b). Categorize and describe the levels of SA, (c). Categorize, summarize and compare the approaches of SA, (d). Discuss the challenges of SA to identify the new trends, (e). Discuss, summarize and compare the literature related to the SA of code-mixed text.

The paper is organized as follows: Section 2 gives a brief introduction to SA while Section 3 presents the different levels of SA. The approaches of SA are described in Section 4. Section 5 discussed the challenges of SA. Section 6 discusses and summarizes the literature related to code-mixing by identifying and comparing the focused

Manuscript received November 5, 2024
Manuscript revised November 20, 2024
https://doi.org/10.22937/IJCSNS.2024.24.11.9

areas, language pairs, datasets, techniques, limitations, and performances. Finally, the conclusion is presented in Section 7.

2. Sentiment Analysis

To apply SA, a sentence needs to be subjective where it contains nonfactual information such as attitudes or opinions. For example "It is sunny" is specified as objective and it conveys a fact or general information. Whereas “I am happy that it is sunny” is subjective and conveys a positive opinion. The opinion or attitude expressed is known as the sentiment and is usually categorized as positive, negative, or neutral according to the polarity value in the range of [-1, 1] where the polarity values less than zero consider as negative sentiments, equal to zero consider as neutral sentiments and greater than zero consider as positive sentiments [4]. In addition to that, some studies have examined the automatic detection of insults, aggregation, hateful/offensive speeches, or emotions like happiness, frustration, anger, sadness, fear, surprise, etc. [1], [5], [6].

SA is a complex task that involves five stages; Data Collection, Text Preprocessing, Feature Extraction, Feature Selection, and Sentiment Classification [7].

The success of SA relies on the quality and the quantity of the data set. An initial data set can be collected through data sources such as social media, review websites, blogs, forums, or interview transcripts. Data from online sources can be obtained by using Application Programming Interfaces (APIs), Open-source data repositories, crowdsourcing, web scrapping, etc. [7].

The initial data sets are user-generated hence data are disorganized, different from user to user, and do not have exact words with exact spellings. Therefore, these initial data sets are not suitable for learning and are essential to normalize by applying preprocessing techniques. Data preprocessing or Data cleaning helps to extract meaningful insights from data and it removes the errors and inconsistencies present in the data. The preprocessing steps are depended on the dataset and the type of analysis. The most common preprocessing steps are, tokenization, removing URLs, removing punctuation marks or symbols or numbers, removing multiple character repetitions, removing stop words, lowering text, stemming, lemmatization, removing other language tags, correcting spellings, etc. [8], [9], [10].

The next step feature extraction is considered as the most important step in the SA process since it increases the performance of the sentiment classification. The main objective of this step is to extract the words which contain the sentiment in the text. One of the most commonly used feature extraction techniques is TF-IDF. This is a method that can be used to convert text into a vector form. The Term Frequency (TF) is the number of times a word occurs in a document. Inverse Document Frequency (IDF) increases the weight of important words (even if those rarely occur) but decreases the weight of unimportant words (even if those frequently occur). Hence, the TF-IDF scheme is used to measure the importance of a word in the document. Another feature extraction technique is Bag of Words (BoW) which also used to convert text into vectors. It assigns higher weightage to the frequently occurring words in the document without considering the order, sentence structure, grammatical construction, or importance of the words. Other well-known feature extraction techniques are n-gram, and Parts-of-Speech (PoS) tagging. N-gram is the contiguous sequence of n items in a text. It identifies the neighboring sequences of items in a document [6], [9], [10]. PoS tagging labeled the words into speech categories such as nouns, verbs, articles, adjectives, etc. [7], [8]. In some studies, opinion words, word count and negation terms have been used as features [11], [12].

The extracted features can be irrelevant and redundant and hence need to be filtered out using feature selection techniques. The advantage of feature selection is that it reduces the size of the feature dimension space and increases the accuracy of SA [7].

The last step is sentiment classification which identifies opinions and classifies them as positive, negative, neutral, hate, good, bad, etc. [13]. To identify the opinion, either Machine Learning-based approaches or Lexicon-based approaches can be used. Machine Learning-based approaches train and test the data set to identify the polarity while Lexicon-based approaches use dictionaries. Once the sentiment classification is finished, the results can be evaluated by using indexes including Precision, Recall, Accuracy, and F1-Score [14].

3. Levels of Sentiment Analysis

According to the task, there are three levels of SA as, document-level, sentence-level, and aspect-level [7].

Document-level SA considers the whole document as a basic information unit and identifies the sentiment. For example, document-level SA can identify the overall sentiment in a product/service review. This level is best for documents that are written by a single person and is not suitable for documents that compare multiple entities or contain opposite sentiments [7].

The sentence-level SA identifies the polarity of a sentence. This level involves two phases: Firstly classifying the sentence as subjective or objective and then identifying the sentiment of a subjective sentence as positive, negative, or neutral [7], [1].

Although SA at the previous two levels is important and useful, these levels do not precisely identify the opinions on aspects of the entity. But aspect-level SA performs better-grained analysis as it classifies the sentiment of a specific aspect of entities. For example, the sentence "The film's songs are awesome, but the storyline is poor" commented on two aspects of the movie, songs, and the storyline. The opinion holder has a positive feeling about the songs and a negative feeling about the storyline. The aspect-level classifies these types of sentences and detects the sentiments expressed in each feature separately [7], [1], [15].

The authors in [16] have implemented an aggression annotated dataset for Hindi-English code-mixed text. The annotation is done at the document-level where a complete post, comment, or any unit of discourse has been considered as a document. The annotation has done in three levels, aggression levels, discursive roles, and discursive effects, and achieved the inter-annotator agreement of 72%.

The study [17] has done a SA on Bengali-English code-mixed text using Convolutional Neural Networks (CNNs). Initially, the code-mixed sentences were classified as positive, negative, or neutral. In the second step, sentences were indexed and each word in each sentence was numbered uniquely. Later the indexed words were represented as vectors and directed to the single-layer CNN model. The model achieved the Accuracy of 0.732.

The researchers in [6] have proposed an aspect-based SA and emotion detection approach for restaurant reviews that use Indonesian-English code-mixed text. The study has considered different aspects such as food, price, service, and ambience. They have created two scenarios where in the first scenario the transformation methods are used for multi-label classification in Machine Learning with unigram features. In the second scenario, Deep Learning algorithms have been used with word embedding. The Random Forest (RF) achieved the highest F1-score of 88.4% with Classifier Chain (CC) method for the food aspect and 89.54% with Label Powerset (LP) method for the price aspect. In the service and ambience aspects, Extra Tree Classifier (ET) dominated with 92.65% and 87.1% with LP and CC methods respectively. In the second scenario, Gated Recurrent Unit (GRU) and Bidirectional Long Short-term Memory (BiLSTM) achieved the same F1-score of 88.16% for the food aspect. GRU performed well with an 83.01% of F1-score for the price aspect and BiLSTM achieved the highest F1-score of 89.03% and 84.78% for the service and ambience aspects respectively.

4. Approaches of Sentiment Analysis

Literature divides the approaches of SA into two categories as Machine Learning-based and Lexicon-based [18].

4.1 Machine Learning-based Approach

Machine Learning-based approaches train and test datasets to identify the sentiment polarity. It is capable of identifying domain-specific patterns and creating models for specific contexts [7]. Machine Learning-based approaches work well with multilingual data (data that consist of multiple languages) as the data set can be trained by using Machine Learning algorithms to classify the sentiments. The success of Machine Learning approaches relies on the quality and the quantity of the dataset [19], [20], [3]. The main drawback of this approach is that a trained classifier just works well with the particular data set only. The same classifier cannot adapt to a new dataset or a new domain [7]. The approach can be divided into three parts, Supervised Learning, Unsupervised Learning, and Semi-supervised Learning [18].

Supervised learning is the most widely used method in SA which trains the classifiers using a labeled corpus with a finite set of classes such as positive, negative, or neutral [18]. The most commonly used supervised learning classifiers are Support Vector Machine (SVM), Artificial Neural Network (ANN), Naïve Bayes (NB), Bayesian Network (BN), and Maximum Entropy (ME) [7].

Authors in [9] have done a study for analyzing the sentiment of agriculture-related comments written in English-Punjabi code-mixed text. The extracted comments were classified sentence-wise as positive, negative, or neutral. Features such as the number of words that match with English-Punjabi sentiment words, the number of ill words, the number of character repetitions, and n-grams have been used. SVM and NB algorithms were used to train the model. The research initially tested the pipeline using a unigram predictive model and later by using n-grams. It has been identified that the performance was enhanced with the n-gram model.

Since supervised learning needs a labeled corpus for training, the data need to be collected and annotated. This is a difficult, time-consuming, and labor-intensive process especially when the text is unstructured. In such situations, unsupervised learning methods can be used since it does not need any prior training with a labeled corpus. Unsupervised learning algorithms are able to find the hidden patterns from a given dataset by themselves without any guidance. Usually unsupervised learning uses statistical approaches or clustering algorithms [7], [15].

An unsupervised SA has been done in [21] for the Spanish-English code-mixed text. The study has implemented methods that use multilingual and cross-lingual embeddings that transfer knowledge from monolingual text to code-mixed text for analyzing the sentiments. This has produced a way to analyze the code-mixed text in a zero-shot way. The study achieved an F1-score of 0.58 without parallel corpus and of 0.62 with parallel corpus on the same benchmark in a zero-shot way.

Semi-supervised learning is used in similar situations as unsupervised learning where it is difficult to acquire a labeled data set. But this method is different from unsupervised learning as it initially needs a small labeled data set for training. Hence the method fits into both supervised and unsupervised methods. Semi-supervised learning uses less amount of data for training and a large volume of data for testing. Most of the Machine Learning problems fall under semi-supervised learning. This method saves time by making use of more unlabeled data and it is even able to create more generalized classifiers [7], [1].

The study [22] has implemented a multilingual semi-supervised approach to detect the polarity in Singaporean English (Singlish) text. For constructing an annotated data set, the study has applied corpus-based bootstrapping using a multilingual, multifaceted lexicon. For identifying the polarity of Singlish n-grams, unsupervised methods such as lexicon polarity detection, frequent item extraction through association rules, and latent semantic analysis have been used. The study has proposed a Singlish polarity detection algorithm and created a hybrid approach by combining the algorithm with an SVM classifier. This hybrid approach achieved the F-measure of 0.78.

4.2 Lexicon-based Approach

In the Lexicon-based method instead of training, a lexicon will be used to identify the polarity values. Lexicon is a predefined list of words where each word is associated with the sentiment polarity. The overall sentiment of a document or sentence is calculated by using the sentiment polarity values of the words that compose it [7]. This approach is more suitable for monolingual data (data that consist of a single language) as the standard lexicons are available. One of the problems associated with this approach is domain dependency. For example, the word "unpredictable" is used in two sentences, "The movie was unpredictable" and "The steering of the car is unpredictable". In the first sentence, the word "unpredictable" express a positive sentiment while in the second sentence the word expresses a negative sentiment. Hence a word can have different senses according to the domain thus a positive word in a specific domain may be a negative word in another domain. This challenge can be handled by using a domain-specific sentiment lexicon [7], [15]. The other problem is that compared to the Machine Learning-based approach, the performance of the Lexicon-based approach is lower when a large data set is used [7].

The paper [10] performed a SA on agriculture-related comments written in English-Punjabi code-mixed text using a statistical technique. The study has created a dictionary of English-Punjabi code-mixed text and categorized the words into three types as positive, negative, and neutral by assigning the polarity values ranging from [-1, 1]. A statistical technique has been used on the dictionary-based dataset at the sentence-level and achieved the highest Accuracy of 83% with the trigrams approach. A SA has been done for the Indonesian and Javanese code-mixed languages in [23] using a Lexicon-based approach. The study has used two lexicons SentiNetWord and VADER to extract the polarity values for the code-mixed text. According to the overall performance, VADER showed better results compared to SentiNetWord.

5. Challenges of Sentiment Analysis

5.1 Sarcasm Detection

Sarcasm means saying or writing something that means the opposite of what it seems to say. Sarcasm is usually used in a humorous way to mock or insult someone. For example, the sentence "Nice perfume, you must shower in it" includes words with a positive opinion. But actually, the sentence express a negative opinion. In these types of sentences, it should identify the actual meaning rather than detect the syntaxes. Hence, the difficulty and ambiguity of sarcasm make SA a very challenging task [7], [14], [15].

The study [24] proposed a Deep Learning based approach to detect sarcasm in Hindi-English code-mixed text. The authors have used two-word embedding approaches, Word2Vec and FastText. They have experimented with different Deep Learning models including CNN, Long Short-term Memory (LSTM), and BiLSTM (with and without attention), and achieved the highest Accuracy of 78.49% with attention-based BiLSTM. The authors in [25] have created the first English-Hindi code-mixed data set for sarcasm detection and also experimented with the data set using three Machine Learning classifiers and 10-fold cross-validation. The study achieved the highest F-score of 78.4 with the RF classifier.

5.2 Ambiguity

The ambiguity can be divided into two parts, Structural Ambiguity (Syntactic Ambiguity) and Lexical Ambiguity (Semantic Ambiguity) [26].

Structural ambiguity results from the different meanings of a sentence [26]. Here the sequence of words is similar but the sentence can be interpreted differently as the sentence may have different syntactic structures in different situations. For example the sentence “The man saw a girl with the telescope” can have two meanings, "The man saw a girl carrying a telescope” or “The man saw a girl through his telescope”.

Lexical ambiguity results from the multiple meanings of a word. For example, the word "Bank" can have two meanings, "a land alongside or sloping down to a river or lake" or "a financial establishment". It is a challenging task for computers to determine the exact meaning of a word according to the particular context. Solving lexical ambiguity is known as Word Sense Disambiguation (WSD) [26].

The paper [2] has implemented a language detection model for the Sinhala-English code-mixed text. The study tried to handle the ambiguity issues presented in the text and noticed that some of the English words, such as "shape", and "royal" are having completely different meanings when it is used in Sinhala-English code-mixing. In addition to that, Sri Lankans usually use "k" to represent the English word "okay". Further, people tend to use "k" at the end of numbers such as “100k” where the value expressed is 100, not 100000. The authors noticed that these kinds of ambiguous words make the SA for the Sinhala-English code-mixed data a complex task as it is difficult to identify the type of language and the appropriate meaning of a particular word. However, the study was able to label some of the ambiguous words such as "royal", and "100k" with the Conditional Random Field (CRF) model. A classification model on Hindi-English code-mixed puns has been implemented in [27] using a four steps process. In the first two steps, the language of each word has been recognized and candidate pun locations have been identified. In the third step, the left and right contexts of the candidate pun locations are looked up, and identified all the possible words which may occur at the location. In the final step, the study calculated the similarity between the words at the location with all the possible words and took the most similar words. This four-step model was able to recover 67% of puns.

5.3 Low-resource Languages

SA is considered as an almost solved problem for a language like English for which a large number of linguistic resources are available. But for the languages like Sinhala and Bambara, linguistic resources are scarce [2], [20]. Most of the SA researches are based on supervised learning approaches which are highly reliant on linguistic resources. Therefore, it is extremely costly to apply supervised learning approaches to the low resources languages. However, using unsupervised or semi-supervised approaches, or constructing linguistic resources from scratch would help to overcome the challenge [7], [15].

The study [20] has done a SA on code-mixed Bambara-French text. They have proposed six Deep Learning models, four LSTM-based models, and two CNN-based models. Since Bambara is a low-resource language, the study used dictionaries of character and word indexes to produce character and word embedding in place of pre-trained word vectors. The study has achieved the highest Accuracy of 83.23% with the one-layer CNN Deep Learning model. The authors [2] presented the first language detection model to detect Sinhala-English code-mixed text. The dataset was newly built by scrapping Facebook chats and posts since this was a novel approach. Manual annotation is done in two phases, annotating sentences to identify the code-mixed text and annotating each word of code-mixed text with language tags. The study was able to develop an XGBoost (XGB) model with 92.1% of Accuracy and a CRF model with an F1-score of 0.94 for sequence labeling.

5.4 Domain Dependency

When using opinion words as a feature, it is necessary to consider the domain since the sentiment polarity can be different according to the domain. For example, the word "fast" is recognized as a negative word in the teaching domain, but it is expressed as a positive sentiment in the phone domain [19]. Therefore, the domain or the context needs to be considered when the sentiment polarity is calculated [15], [14].

The study [12] has done a domain-specific SA for the Hindi-English code-mixed text. They have proposed a hybrid system that incorporates both Lexicon-based approaches and Machine Learning-based approaches. In the Lexicon-based approach, a lexicon that represented the movie domain has been created and the lexicon contained a list of slang and abbreviated words in both languages. The Lexicon-based approach achieved the highest Accuracy of 86% and the Machine Learning-based approach achieved the highest Accuracy of 72%. A Named Entity Recognition (NER) for two domains, sports and tourism has been done in the study [28] for Bengali-English code-mixed text. The study has proposed two different NE taxonomies separately for each domain. Classes and features are also employed according to the domain. The experiments have been done with four Machine Learning classifiers including CRF, Margin Infused Relaxed Algorithm (MIRA), SVM, and Maximum Entropy Markov Model (MEMM). The CRF performed well in the sports domain and SVM achieved the highest accuracy in the tourism domain.

6. Comparison of Code-Mixed Text Literature

According to the literature found it was identified that most of the code-mixed text related researches are focused on four areas including (a). Preprocessing, (b). Language identification, (c). Corpus creation, (d). Sentiment or Emotion classification [18].

6.1 Preprocessing

The studies on preprocessing were mainly focused on tasks such as noisy text identification, spell correction, and stop words removal [18]. The authors in [29] have done a study to correct the misspelled English words in

Bangla-English code-mixed text through word-level language identification. The identified English words which do not appear in the vocabulary have been considered as misspelled words and directed to a spell checker. The spell checker is based on the noisy channel model and it tackled wordplay, contracted words, and phonetic variations. The spell checker obtained the Accuracy of 69.43%. The study [30] has proposed a pipeline model to normalize the Indonesian-English code-mixed tweets using four modules, tokenization, language identification, lexical normalization, and translation. In the first two modules, the tweets were tokenized and all the tokens were tagged with the corresponding language tags. In the lexical normalization module, each token was taken as an input with the language tags and mapped with their standard formats using word distribution along with the rule-based method. The last module merged the normalized tokens back into the tweet and translated them into Indonesian. The pipeline achieved a score of 54.07 for Bilingual Evaluation Understudy (BLEU) and a score of 31.89 for Word Error Rate (WER).

6.2 Language Identification

Language identification is considered as a challenging task in social media code-mixed context. The study [31] has done a word-level language identification for the Bengali-English code-mixed text. The study has built two LSTM models using character encoding and phonetic encoding. By combining these two models, the study has implemented two ensemble models using the stack and threshold techniques. The stacking model achieved the Accuracy of 91.78% and the threshold model achieved the Accuracy of 92.35%. The study [32] also did a word-level language identification for the Turkish-Dutch code-mixed text. They have used dictionary lookup, language model, and dictionary + language model as features and implemented Logistic Regression (LR) and CRF-based models. The study identified that language models are more robust than dictionaries and performance can be improved by considering the context. The study [33] also proposed a feature-based embedded methodology to identify the language tags at the word-level for Sinhala-English code-mixed sentences. The study achieved the highest Accuracy of 90.5% with the RF classifier. Authors in [8] experimented with different models for language identification in English-Telugu code-mixed data. The best output has been given by the CRF model with an F1-score of 0.91. With the CRF model, the study has considered a set of features that were based on different possible combinations of words, context, and possible tags.

6.3 Corpus Creation

Although various corpora are available for monolingual languages such as English, Russian, Norwegian, and Hindi, there is a limited number of corpus and lexicon resources exist for code-mixed languages. Therefore, corpus creation is one of the significant tasks in code-mixed-based researches. The study [34] has created an annotated Tamil-English code-mixed corpus with 15,744 comments. The comments have been collected using the YouTube comment scraper tool and filtered out the noncode-mixed comments using the langdetect library. The study used Krippendorff's alpha to measure the inter-annotator agreement and achieved the agreement of 0.6585 using nominal metric and 0.6799 using interval metric. The authors in [35] have prepared a Bengali-English code-mixed corpus using two phases of annotation such as language tagging, and sentiment tagging. The study achieved the inter-annotator agreement of 0.83 for language tagging and 0.94 for sentiment tagging. The corpus has been used for the classification of several features. The language tagger achieved the Accuracy of 81% and the sentiment tagger achieved the Accuracy of 80.97%. The authors in [11] created a corpus and identify the hate speeches in Hindi-English code-mixed text using 4574 tweets. The annotation has been done in two phases, word-level language annotation and hate speech annotation, and achieved the kappa value of 0.982 for the hate speech annotation. The study [36] has created a code-mixed corpus for the language pair, Malayalam-English with an inter-annotator agreement of 0.8. The corpus contained 6739 comments on movie trailer reviews. In the annotation process, they have identified some ambiguity issues such as commentators comparing a movie with some other movies and commenting on the different aspects of the movie in the same sentence. This makes it difficult to identify the actual sentiment expressed by the viewer.

6.4 Sentiment or Emotion Classification

The purpose of sentiment or emotion classification is to identify the sentiment or emotion expressed in a text and label them as positive, negative, neutral, hate, happy, etc. The paper [37] implemented a model to detect hate speeches in Hindi-English code-mixed text. The study has used Facebook's pre-trained library fastText to identify speeches. The proposed model has been compared with word2vec and doc2vec algorithms and identified that the performance of the implemented model is high. And also they observed that in the code-mixed classification, character-level features give more details compared to the word and document-level features. The study [38] has done an emotion detection for the Hindi-English code-mixed text by creating a corpus with 12000 texts. They used three classes and maintained an equal number of texts for each class to omit the class imbalanced problem. A bilingual pre-trained model has been retrained using Word2Vec to convert texts into vectors. Different Deep Learning models have been used and CNN-BiLSTM achieved the highest Accuracy of 83.21%.

Table 1 summarizes the literature related to code-mixing by comparing the language pairs, datasets, techniques, limitations, and performances.

Table 1: Comparison between Code-Mixed based Studies

Language Pair	Ref.	Source of the data set	Task	Approach/ Algorithms	Performance/ Results	Limitations/ Challenges
Bambara-French	[20]	Facebook	SA	One-layer LSTM, Two-layer LSTM. One-layer BLSTM, Two-layer BLSTM, One-layer CNN, CNN-LSTM, SVM, NB	Best Accuracy of 83.23% for One-layer CNN	-
Bengali-English	[35]	Twitter	Corpus creation	-	Kappa value=0.83 for language tag Kappa value=0.94 for sentiment tag	-
	[17]	ICON17[1]	SA	CNN	Accuracy=0.732	-
	[29]	Social media	Text normalization	Language identification: CRF, and Post-processing heuristics. Spell checking: Noisy channel model	Highest Accuracy of 90.50% for Language identification Highest Accuracy of 69.43% for the Spell checker	Unable to handle misspelled words with more than two errors and words with punctuation marks
	[31]	ICON16[2], ICON17, and the dataset used in [39]	Language identification	Two LSTM models based on character encoding and phonetic encoding	Accuracy of 91.78% with the stacking method Accuracy of 92.35% with threshold method	Failed to capture the context information, Unable to handle elongated words and words with numeric or special characters
	[28]	Dataset described in [40]	NER	CRF, MIRA, SVM, and MEMM	CRF performed best in the sports domain SVM performed well in the tourism domain	Artifacts were misclassified as locations, Four digits numbers were misclassified, Incorrectly tagged some words such as "temple", "river", and "taxi", and Some sports-related words were misclassified.
Indonesian and Javanese	[23]	Twitter	SA	A Lexicon-based approach	VADER showed better results compared to SentiNetWord.	Comparatively low performance with positive and neutral sentiment classification
English-Punjabi	[9]	Twitter, Facebook, and YouTube	SA	SVM and NB	NB achieved the highest Accuracy with tri-gram	-

[1] https://ltrc.iiit.ac.in/icon2017/

[2] http://ltrc.iiit.ac.in/icon2016/

Language Pair	Ref.	Source of the data set	Task	Approach/ Algorithms	Performance/ Results	Limitations/ Challenges
	[10]	Twitter, Facebook, and YouTube	SA	A statistical technique	Accuracy of 83% with the trigrams approach	-
English-Spanish	[21]	Datasets provided by [41], Sentistrength[1], [42], [43], and [44]	SA	LSTM	F1-score of 0.58 without parallel corpus and 0.62 with parallel corpus	-
Turkish-Dutch	[32]	An online forum	Language Identification	LR, CRF	Language models are more robust than dictionaries and adding context improves the performance.	Words containing both Turkish and Dutch spellings didn't classify properly, System ignored the named entities
English-Telugu	[8]	ICON2015[2]	Language Identification	NB, RF, Hidden Markov (HM), CRF	CRF with the Accuracy of 91.28%.	Faced challenges with spelling errors and SMS-type conversations
Sinhala-English	[2]	Facebook	Language Identification	NB, LR, SVM, RF, XGB, shallow Neural Network, Deep Neural Network, LSTM, CNN, recurrent CNN, bidirectional Recurrent Neural Network, Gated Recurrent Unit, CRF, and K-Nearest Neighbor (K-NN)	Classification: XGB with 92.1% of Accuracy Sequence labeling: CRF with an F1-score of 0.94	Insufficient data set
	[33]	Facebook	Language Identification	SVM, NB, LR RF, and Decision Tree (DT)	RF with an Accuracy of 90.5%	Didn't identify the 'rest' tags (named entities, acronyms, and other language tags) accurately
Malayalam-English	[36]	YouTube	Corpus Creation	-	Krippendorff's alpha= 0.890	Difficult to annotate comments which made comparisons in-between movies and different aspects of movies
Tamil-English	[34]	YouTube	Corpus Creation and SA	LR, SVM, K-NN, DT, RF, Multinominal Naive Bayes, 1DConv-LSTM, Dynamic Meta Embedding, BERT-Multilingual	Agreement= 0.6585 using nominal metric and Agreement=0.6799 using interval metric	The corpus is imbalanced

[1] http://sentistrength.wlv.ac.uk/

[2] https://ltrc.iiit.ac.in/icon2015/

Language Pair	Ref.	Source of the data set	Task	Approach/ Algorithms	Performance/ Results	Limitations/ Challenges
Indonesian-English	[6]	PergiKuliner platform	SA	Machine Learning algorithms: DT, RF, SVM, ET Deep Learning algorithms: BiLSTM, GRU	Machine Learning algorithms: Food aspect-RF with F1-score of 88.4%, Price aspect-RF with F1-score of 89.54%, Service aspect-ET with F1-score of 92.65%, Ambience aspect-ET with F1-score of 87.1% Deep Learning algorithms: Food aspect- GRU and BiLSTM with F1-score of 88.16%, Price aspect-GRU with F1-score of 83.01%, Service aspect-BiLSTM with F1-score of 89.03%, Ambience aspect-BiLSTM with F1-score of 84.78%	Imbalanced data set
	[30]	Twitter and datasets used in [45] and [46]	Normalization	A pipeline with four modules	A score of 54.07 for BLEU and a Score of 31.89 for WER	The performance of the translation module is low
Singaporean English (Singlish)	[22]	Twitter	Polarity Detection	Semi-supervised approach	A hybrid approach achieved the F-measure of 0.78	The presence of ambiguous words reduced the accuracy
Hindi-English	[25]	Twitter	Sarcasm Detection	SVM-RBF,SVM-linear, and RF	RF classifier achieved the best score of 78.4	-
	[11]	Twitter	Corpus Creation and Hate Speech Detection	SVM, RF	Kappa value= 0.982, SVM achieved the best Accuracy of 71.7%	-
	[16]	Facebook and Twitter	Corpus Creation	-	Inter annotator Agreement for the top level is above 72% and for the 10-class annotation is 57%	-
	[38]	Data Set used in [11], Twitter, Facebook, and Instagram	Emotion Detection	ID-CNN, LSTM, Bi-LSTM, CNN-LSTM, CNN-BiLSTM	CNN-BiLSTM achieved the best Accuracy of 83.21%	-
	[37]	Data sets used in [11], [47] and Hate Speech and Offensive Content Identification in Indo-European Languages	Hate Speech Detection	SVM-linear, SVM–Radial Basis Function (RBF), and RF	The best feature representation was given by SVM-RBF with fastText features	-

Language Pair	Ref.	Source of the data set	Task	Approach/ Algorithms	Performance/ Results	Limitations/ Challenges
		(HASOC)				
	[24]	Twitter	Sarcasm Detection	CNN, LSTM, BiLSTM (with and without attention)	Attention-based BiLSTM achieved the highest Accuracy of 78.49%	Lack of clean data, an insufficient data set
	[27]	Advertisements	Recover puns	A four-step model which used language models, and phonetic similarity-based features	Recover 67% of puns	The model fails when a pun translates to multiple words, when puns are based on the pronunciations of abbreviations, or bigram does not exist in puns
	[12]	Facebook	SA	Lexicon-based approach	86%	-
				NB, SVM, DT, Random Tree (RT), Multilayer Perceptron	72%	

7. Conclusion

Most social media users mix two or more languages or language varieties in speech. This situation is known as code-mixing. SA of code-mixed text is a challenging task from the data collection to the sentiment classification since the text contains informal grammar, spelling variations, creative spelling, nonstandard abbreviations, undetermined mixing rules, and noise. However, with the advancement of NLP tools and techniques, code-mixed text-based studies have gained huge attention.

This paper has attempted to study the literature on state-of-the-art in SA of code-mixed text. The general process, levels, basic approaches, and challenges of SA of code-mixed text have been highlighted in the paper. The paper also discussed, summarized, and compared language pairs, datasets, tasks, approaches, performances, and limitations of various code-mixed-based studies.

It was identified that feature extraction is the most important step in the SA. Feature extraction has a direct impact on the performance of sentiment classification since it extracts valuable information about the characteristics of the text. The study shows that there are three levels of SA and among them, aspect-level SA is more challenging and interesting as it is able to identify the sentiments of each aspect of entities. The study also reveals that Machine Learning-based approaches are more suitable for multilingual or code-mixed text as code-mixed text often do not contain standard lexicons. However, the performance of Machine Learning approaches depends on the quality and the quantity of the dataset. The study was able to identify many challenges faced by SA of code-mixed text and among them, the main challenge identified was sarcasm detection.

According to the literature found it was noticed that SA studies for code-mixed text are mainly centered on preprocessing, language identification, corpus creation, and sentiment classification tasks. Due to the lack of linguistic resources available, the researches based on SA of code-mixed text are still at the beginning for some language pairs such as Bambara-French, and Sinhala-English. But, a huge interest can be observed in Indian language pairs including Hindi-English, and Bengali-English. The study reveals that Facebook and Twitter are the most common data collection sources. The literature shows that from the traditional Machine Learning classifiers, RF followed by SVM and CRF achieved the highest Accuracies and F1-scores. From Neural Network approaches, BiLSTM (BLSTM) followed by CNN performed well. When the data set is huge, Neural Network approaches perform better than traditional Machine Learning approaches. It was observed that most of the studies related to the SA of code-mixed text have faced challenges with spelling errors and limited datasets hence some tags were misclassified and achieved low performances.

References

[1] G. I. Ahmad, J. Singla, and N. Nikita, “Review on Sentiment Analysis of Indian Languages with a Special Focus on Code Mixed Indian Languages,” in 2019 International Conference on Automation, Computational and Technology Management (ICACTM), pp. 352–356, Apr. 2019.

[2] I. Smith and U. Thayasivam, “Language Detection in Sinhala-English Code-mixed Data,” in 2019 International Conference on Asian Language Processing (IALP), pp. 228–233, Nov. 2019.

[3] R. Srinivasan and C. N. Subalalitha, “Sentimental analysis from imbalanced code-mixed data using machine learning approaches,” Distrib. Parallel Databases, Mar. 2021.

[4] P. Mishra, P. Danda, and P. Dhakras, “Code-Mixed Sentiment Analysis Using Machine Learning and Neural Network Approaches,” arXiv, arXiv:1808.03299, Aug. 2018.

[5] G. Kovács, P. Alonso, and R. Saini, “Challenges of Hate Speech Detection in Social Media,” SN Comput. Sci., vol. 2, no. 2, p. 95, Feb. 2021.

[6] A. Suciati and I. Budi, “Aspect-Based Sentiment Analysis and Emotion Detection for Code-Mixed Review,” Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 9, 2020.

[7] M. Birjali, M. Kasri, and A. Beni-Hssane, “A comprehensive survey on sentiment analysis: Approaches, challenges and trends,” Knowl.-Based Syst., vol. 226, p. 107134, Aug. 2021.

[8] S. Gundapu and R. Mamidi, “Word Level Language Identification in English Telugu Code Mixed Data,” arXiv, arXiv:2010.04482, Oct. 2020.

[9] M. Singh, V. Goyal, and S. Raj, “Sentiment Analysis of English-Punjabi Code Mixed Social Media Content for Agriculture Domain,” in 2019 4th International Conference on Information Systems and Computer Networks (ISCON), pp. 352–357, Nov. 2019.

[10] Mukhtiar Singh, Vishal Goyal, and Sahil Raj, “Sentiment Analysis of Social Media Tweets on Farmer Bills 2020,” J. Sci. Res., vol. 65, no. 3, 2021.

[11] A. Bohra, D. Vijay, V. Singh, S. S. Akhtar, and M. Shrivastava, “A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection,” in Proceedings of the Second Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media, New Orleans, Louisiana, USA, pp. 36–41, Jun. 2018.

[12] A. Pravalika, V. Oza, N. P. Meghana, and S. S. Kamath, “Domain-specific sentiment analysis approaches for code-mixed social network data,” in 8th International Conference on Computing, Communication, and Networking Technologies (ICCCNT), pp. 1–6, 2017.

[13] A. D’Andrea, P. Grifoni, and Guzzo, “Approaches, Tools and Applications for Sentiment Analysis Implementation,” Int. J. Comput. Appl., vol. 125, no. 3, 2015.

[14] V. A. Kharde and P. S. Sonawane, “Sentiment Analysis of Twitter Data: A Survey of Techniques,” Int. J. Comput. Appl., vol. 139, no. 11, pp. 5–15, Apr. 2016.

[15] M. Joshi, P. Prajapati, A. Shaikh, and V. Vala, “A Survey on Sentiment Analysis,” Int. J. Comput. Appl., vol. 163, pp. 34–38, Apr. 2017.

[16] R. Kumar, A. N. Reganti, A. Bhatia, and T. Maheshwari, “Aggression-annotated Corpus of Hindi-English Code-mixed Data,” arXiv, arXiv:1803.09402, Mar. 2018.

[17] S. K., A. Ravikurnar, V. R C., A. Reddy D., A. K. M., and S. K.P., “Sentiment Analysis of Indian Languages using Convolutional Neural Networks,” in 2018 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–4, Jan. 2018.

[18] Nurul Husna Mahadzir, Mohd Faizal Omar, Mohd Nasrun Mohd Nawi, Anas A. Salameh, and Kasmaruddin Che Hussin, “Sentiment Analysis of Code-Mixed Text: A Review,” Turk. J. Comput. Math. Educ., vol. 12, no. 3, pp. 2469–2478, 2021.

[19] B. Sharounthan, D. P. Nawinna, and R. De Silva, “Singlish Sentiment Analysis Based Rating For Public Transportation,” in 2021 International Conference on Computer Communication and Informatics (ICCCI), pp. 1–7, Jan. 2021.

[20] A. Konate and R. Du, “Sentiment Analysis of Code-Mixed Bambara-French Social Media Text Using Deep Learning Techniques,” Wuhan Univ. J. Nat. Sci., vol. 23, no. 3, pp. 237–243, Jun. 2018.

[21] S. Yadav and T. Chakraborty, “Unsupervised Sentiment Analysis for Code-mixed Data.” arXiv, Jan. 20, 2020.

[22] S. L. Lo, E. Cambria, R. Chiong, and D. Cornforth, “A multilingual semi-supervised approach in deriving Singlish sentic patterns for polarity detection,” Knowl.-Based Syst., vol. 105, pp. 236–247, Aug. 2016.

[23] C. Tho, Y. Heryadi, L. Lukas, and A. Wibowo, “Code-mixed sentiment analysis of Indonesian language and Javanese language using Lexicon based approach,” J. Phys. Conf. Ser., vol. 1869, no. 1, p. 012084, Apr. 2021.

[24] A. Aggarwal, A. Wadhawan, A. Chaudhary, and K. Maurya, “‘Did you really mean what you said?’ : Sarcasm Detection in Hindi-English Code-Mixed Data using Bilingual Word Embeddings,” in Proceedings of the Sixth Workshop on Noisy User-generated Text (W-NUT 2020), pp. 7–15,2020.

[25] S. Swami, A. Khandelwal, V. Singh, S. S. Akhtar, and M. Shrivastava, “A Corpus of English-Hindi Code-Mixed Tweets for Sarcasm Detection.” arXiv, May 30, 2018.

[26] J. Arukgoda, V. Bandara, S. Bashani, V. Gamage, and D. Wimalasuriya, “A Word Sense Disambiguation Technique for Sinhala,” in 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology, pp. 207–211, Dec. 2014.

[27] S. Aggarwal, K. Mathur, and R. Mamidi, “Automatic Target Recovery for Hindi-English Code Mixed Puns.” arXiv, Jun. 11, 2018.

[28] Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay, “Named Entity Recognition on Code-Mixed Cross-Script Social Media Content,” vol. 21, no. 4, pp. 681–692, 2017.

[29] S. Dutta, T. Saha, S. Banerjee, and S. K. Naskar, “Text normalization in code-mixed social media text,” in 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS), pp. 378–382, Jul. 2015.

[30] A. M. Barik, R. Mahendra, and M. Adriani, “Normalization of Indonesian-English Code-Mixed

Twitter Data," in Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019), Hong Kong, China, pp. 417–424, Nov. 2019.

[31] S. Mandal, S. D. Das, and D. Das, "Language Identification of Bengali-English Code-Mixed data using Character & Phonetic based LSTM Models," arXiv, arXiv:1803.03859, Jun. 2018.

[32] D. Nguyen and A. S. Doğruöz, "Word Level Language Identification in Online Multilingual Communication," in Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, Washington, USA, pp. 857–862, Oct. 2013.

[33] K. Shanmugalingam and S. Sumathipala, "Language identification at word level in Sinhala-English code-mixed social media text," in 2019 International Research Conference on Smart Computing and Systems Engineering (SCSE), pp. 113–118, Mar. 2019.

[34] B. R. Chakravarthi, V. Muralidaran, R. Priyadharshini, and J. P. McCrae, "Corpus Creation for Sentiment Analysis in Code-Mixed Tamil-English Text," arXiv, arXiv:2006.00206, May 2020.

[35] S. Mandal, S. K. Mahata, and D. Das, "Preparing Bengali-English Code-Mixed Corpus for Sentiment Analysis of Indian Languages," arXiv, arXiv:1803.04000, Mar. 2018.

[36] B. R. Chakravarthi, N. Jose, S. Suryawanshi, E. Sherly, and J. P. McCrae, "A Sentiment Analysis Dataset for Code-Mixed Malayalam-English," Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL). European Language Resources Association (ELRA), Marseille, France, pp. 177–184, May 30, 2020.

[37] K. Sreelakshmi, B. Premjith, and K. P. Soman, "Detection of Hate Speech Text in Hindi-English Code-mixed Data," in Third International Conference on Computing and Network Communications (CoCoNet'19), vol. 171, pp. 737–744, 2020.

[38] T. T. Sasidhar, P. B, and S. K. P, "Emotion Detection in Hinglish(Hindi+English) Code-Mixed Social Media Text," Procedia Comput. Sci., vol. 171, pp. 1346–1352, 2020.

[39] S. Mandal and D. Das, "Analyzing Roles of Classifiers and Code-Mixed factors for Sentiment Identification." arXiv, Mar. 15, 2018.

[40] Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay, "The First Cross-Script Code-Mixed Question Answering Corpus," in MultiLingMine@ ECIR, 2016.

[41] S. Rosenthal, N. Farra, and P. Nakov, "SemEval-2017 Task 4: Sentiment Analysis in Twitter," in Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Vancouver, Canada, pp. 502–518, Aug. 2017.

[42] J. A. Cerón-Guzmán and S. de Cali, "Classifier Ensembles That Push the State-ofthe-Art in Sentiment Analysis of Spanish Tweets," 2017.

[43] J. V. Roman, E. M. Camara, and J. G. Morera, "TASS 2014 - The Challenge of Aspect-based Sentiment Analysis," in Procesamiento de Lenguaje Natural, pp. 61–68, 2015.

[44] D. Vilares, M. A. Alonso, and C. Gómez-Rodríguez, "EN-ES-CS: An English-Spanish Code-Switching Twitter Corpus for Multilingual Sentiment Analysis," in Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), Portorož, Slovenia, pp. 4149–4153, May 2016.

[45] A. Go, R. Bhayani, and L. Huang, "Twitter Sentiment Classification using Distant Supervision," Stanford, CS224N Project Report, 2009.

[46] Putra Pandu Adikara, "Normalisasi kata pada pesan/status singkat berbahasa indonesia," Universitas Indonesia, Depok, 2015.

[47] P. Mathur, R. Sawhney, M. Ayyar, and R. Shah, "Did you offend me? Classification of Offensive Tweets in Hinglish Language," in Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), Brussels, Belgium, pp. 138–148 Oct. 2018.

W.A.S.C. Perera received the B.Sc. in ICT from University of Jaffna in 2014 and M.Sc. in Computer Science from University of Colombo in 2019. After working as a Temporary Lecturer (from 2014-2016) in the Dept. of Physical Science, Vavuniya Campus, she has been a Lecturer at Dept. of ICT, University of Vavuniya since 2017. Her research interest includes Data Mining, Machine Learning, Natural Language Processing, and Computational Linguistics.

H.A. Caldera received his PhD degree in computer science from University of Western Sydney, Australia in 2005. Since then, he has been a Senior Lecturer at the University of Colombo School of Computing. His current interests include Data Mining, Web Mining, Business Intelligence and Natural Language Processing. He has published and reviewed many international conferences and journal papers.