Journal of Universal Computer Science, vol. 30, no. 2 (2024), 242-261
submitted: 11/12/2022, accepted: 26/9/2023, appeared: 28/2/2024 CC BY-ND 4.0

Sentiment Analysis of Code-Mixed Text: A Comprehensive Review

Anne Perera
(Faculty of Technological Studies, University of Vavuniya, Vavuniya, Sri Lanka
https://orcid.org/0000-0003-1940-9132, anneshehari@gmail.com)

Amitha Caldera
(University of Colombo School of Computing, Colombo 07, Sri Lanka
https://orcid.org/0000-0003-0288-536X, hac@ucsc.cmb.ac.lk)

Abstract: Sentiment Analysis is the task of identifying and extracting the opinion expressed in a text to determine the writer's perception of an entity. Due to globalization, people often mix two or more languages and use phonetic typing and lexical borrowing in web communication. This concept is known as code-mixing. Although extracting the opinion of text written in monolingual languages is simple and straightforward, Sentiment Analysis of code-mixed text is challenging. Classifiers fail within the context of the code-mixed text as text may consist of creative writing, spelling variations, grammatical errors, and different word orders. Hence, SA of code-mixed text is an interesting, challenging, and popular research area. This paper presents the state-of-the-art in Sentiment Analysis of code-mixed text by discussing each concept in detail. The paper also discusses the focused areas, techniques used, limitations, and performances of the studies related to code-mixing.

Keywords: Code-mixed, Monolingual, Natural Language Processing, Sentiment Analysis
Categories: A.1, I.1.7
DOI: 10.3897/jucs.98708

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 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 cannot interpret natural languages. In that case, the Sentiment Analysis (SA) technique can be applied to identify the opinion hidden behind a text using Natural Language Processing (NLP).

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 helpful in social media monitoring, business, politics, and almost all fields since it gains in-depth insight into people's attitudes regarding trends, persons, organizations, products, or services [Ahmad et al. 2019]. A sentence needs to be subjective, where it contains nonfactual information such as attitudes or opinions to apply SA. 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 gives a positive opinion. The opinion or attitude expressed is known as the sentiment. Sentiments are usually categorized as positive, negative, or neutral according to the polarity value in the range of [-1, 1]. The polarity values less than zero are considered negative sentiments, equal to zero are considered neutral sentiments, and greater than zero are considered positive sentiments [Mishra et al. 2018]. In addition, some studies have examined the automatic detection of insults, aggregation, offensive speeches, or emotions like happiness, frustration, anger, sadness, fear, surprise, etc. [Ahmad et al. 2019], [Suciati and Budi 2020], [Kovács et al. 2021].

Internet users are from all over the world; hence, they often bring their language, background, and culture to web communication. Although English is considered 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. This concept is known as code-mixing, and this is the latest trend in web communication. In simple words, code-mixing means mixing two or more languages or language varieties in speech. As a result of code-mixing, different variations of languages have emerged [Smith and Thayasivam 2019].

Although extracting the opinion of text written in English or Unicode characters is straightforward, SA of code-mixed text is challenging. The usual preprocessing techniques used for monolingual SA, such as stemming, Part-of-Speech tagging (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 [Srinivasan and Subalalitha 2023]. As a result, the code-mixed text differs from user to user and does not have exact words with exact spellings as in monolingual languages. Some challenges of code-mixed text are lack of formal grammar, spelling variations, creative spelling, undetermined mixing rules, noise, nonstandard abbreviations, long processes, and lack of linguistic resources available [Ahmad et al. 2019], [Srinivasan and Subalalitha 2023].

SA is a challenging and interesting research field; hence, it has gained enormous attention among the research communities over the last one and a half-decades. A massive increment can be observed in the number of studies focused on SA [Birjali et al. 2021]. The studies focused on code-mixing are limited in number. However, an improvement can be observed in the field with the help of advanced NLP tools and techniques.

Several surveys and reviews have been conducted in the field of code-mixing. The authors [Thara and Poornachandran 2018] conducted a comprehensive study on code-mixing by comparing the works published between 2013 and 2017. The paper presented the applications of code-mixing with brief details. The authors also compared the performances of different NLP approaches in the code-mixed context. The study [Ahmad et al. 2019] reviewed the SA of code-mixed Indian languages. The authors investigated the approaches and issues of SA and presented the availability of linguistic resources for different Indian languages. Furthermore, the study showed that the majority of works had been conducted in Hindi, Bengali, and Tamil languages. The paper [Tho et al. 2020] analyzed studies on classifiers used in SA of code-mixed text. The study provided a comparison between the code-mixed-based studies and identified that the most common classifiers used were Support Vector Machine (SVM), Naïve Bayes (NB), and Logistic Regression (LR). The authors [Mahadzir et al. 2021] presented state-of-the-art SA of code-mixed text. The paper defined the problem of using multilingual languages in web communication and discussed the focused areas and current approaches in detail. The authors identified that the studies focused on code-mixing centered on five tasks: preprocessing, language identification, lexicon creation, sentiment classification, and subjectivity analysis. Furthermore, the study presented a qualitative comparison among existing works' data sets, techniques, and limitations. The paper also highlighted some issues and challenges of code-mixed text analysis. The survey [Ahmad et al. 2022] investigated the Machine Learning classifiers used in SA of code-mixed and code-switched Indian languages. The study identified that the most used classifier was SVM, followed by NB and Random Forest (RF). The study also compared the languages, approaches, data sets, and challenges in the context. They showed that Twitter was the most common data collection method and Hindi-English was the most researched Indian language pair. The paper [Hidayatullah et al. 2022] presented a survey on language identification of code-mixed text. The study focused on techniques, challenges, and data availability. The authors identified 32 code-mixed data sets for language identification and proposed a code-mixed language identification framework as a guideline for future studies.

The main purpose of this review paper is to examine each key concept of SA of code-mixed text in detail and analyze the existing works related to each concept. The paper discusses the concepts and literature of SA of code-mixed text, including the levels, approaches, challenges, performances, and limitations. The main contributions of the study are: (a). Describes the generic process of SA, (b). Categorizes and describes the levels of SA, (c). Categorizes, and compares the approaches of SA, (d). Discusses the challenges of SA to identify the new trends, (e). Discusses the literature related to the SA of code-mixed text.

The rest of the paper is organized as follows: Section 2 describes the methodology that was adopted to identify the relevant existing works. The research findings are presented from Section 3 to Section 9. Section 3 briefly describes the SA process, while Section 4 presents the different levels of SA. The approaches of SA are described in Section 5. Section 6 discusses the challenges of SA. Section 7 discusses the studies related to code-mixing according to the focused areas by identifying language pairs, techniques, limitations, and performances. The discussion and conclusion are presented in Section 8 and Section 9 respectively.

2 Research Methodology

This study adopted the review methodology described in [Keele 2007], [Qazi et al. 2017] for identifying the resources of SA of code-mixed text. The paper applied the following review strategies: (a). developing research questions, (b). searching related papers from electronic databases using search strings, (c) applying inclusion and exclusion criteria, and (d) applying quality assessment criteria.

2.1 Research Questions

The study aims to answer the following research questions:

Question 1: What is the generic process followed in code-mixed text analysis?

Question 1 aims to understand the key steps followed in the existing studies related to the SA of code-mixed text. Understanding the generic process is important as it allows to learn from similar studies and reuse the structure in future studies. The findings can be used as a standard framework for those researching SA of code-mixed text.

Question 2: Which SA levels have been used in code-mixed text analysis?

Question 2 aims to identify and categorize the different levels of SA of code-mixed text. Identifying the levels on which SA of code-mixed text is performed, allows us to understand the most suitable applicable level according to the task and the dataset in future research.

Question 3: Which NLP techniques have been used in code-mixed text analysis?

Question 3 allows us to identify the methods used in the research works related to the SA of code-mixed text. Examining previously used approaches will provide insight into the state-of-the-art, advantages and limitations of SA of code-mixed text. The findings will demonstrate the most recommended techniques for dealing with code-mixed text.

Question 4: What are the challenges of code-mixed text analysis?

Question 4 aims to identify the key challenges that make it difficult to analyze the code-mixed text and detect the sentiment polarities. Understanding the challenges is necessary to determine the research gaps in the existing works that are currently not addressed or answered adequately. The finding would imply the directions for future research.

Question 5: How to categorize the works related to code-mixing by the task?

Question 5 aims to identify and categorize the focused areas of code-mixed text-based studies according to the task. Examining previous studies in each area will provide insight into datasets, performances, and limitations of SA of code-mixed text. The findings will help to recognize the implications for current practices and future research.

2.2 Search Strategy

The related studies were searched in four electronic sources: Google Scholar, IEEE Explore, Science Direct, and Research Gate. Through these databases, the study investigated all available materials related to the objectives of the literature review. Search strings were designed to identify research papers answering the research questions. The strings were “Sentiment Analysis AND code-mixed”, “Machine Learning AND code-mixed”, “Lexicon-based AND code-mixed”, and “Challenges AND Sentiment Analysis AND code-mixed”.

2.3 Inclusion and Exclusion Criteria

Meta-data and the abstracts of the papers were reviewed to determine the relevant articles and remove the irrelevant articles. The following criteria were applied for inclusion: (a). studies published from 2015 to 2022, (b). full-text papers, (c). articles written in English, (d). studies related to SA of code-mixed text. The exclusion criteria were as follows: (a). unrelated papers, (b). unpublished works, (c). studies not focused on NLP. The study excluded the studies that did not satisfy the inclusion criteria or the studies that matched any of the exclusion criteria.

2.4 Quality Assessment Criteria

The quality assessment criteria were used in the study to determine the strength of the selected publications. The quality assessment criteria were as follows: (a). the article described the data set clearly, (b). the article explained the techniques clearly, (c). the article clearly stated the findings.

2.5 Selection Process

The study searched the literature from four electronic databases using four search strings. A total of 259 papers were obtained in the search process, and 136 were identified as duplicates. Next, the inclusion and exclusion criteria were applied to the rest of the 123 papers and a total of 62 papers were eliminated. After that, the rest of the 61 articles were assessed using the quality assessment criteria, and a total of 32 papers were eliminated. Finally, the balance of 29 papers were selected for the literature review.

3 Generic Process of Sentiment Analysis

SA is a complex task that involves five stages; Data Collection, Text Preprocessing, Feature Extraction, Feature Selection, and Sentiment Classification [Singh et al. 2019], [Gundapu and Mamidi 2020], [Birjali et al. 2021].

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 through Application Programming Interfaces (APIs), open-source data repositories, crowdsourcing, web scrapping, etc. [Singh et al. 2019], [Gundapu and Mamidi 2020], [Birjali et al. 2021].

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 unsuitable for learning and essential to normalize by applying preprocessing techniques. Data preprocessing or Data cleaning helps extract meaningful insights from data and removes the errors and inconsistencies in the data. The preprocessing steps depend on the data set 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. [Kharde and Sonawane 2016], [Singh et al. 2019], [Gundapu and Mamidi 2020].

The next step, feature extraction, is considered the most crucial step in the SA process since it increases the sentiment classification performance. 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. TF-IDF is a method that converts 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 is also used to convert text into vectors. It assigns higher weightage to the frequently occurring words in the document without considering the words' order, sentence structure, grammatical construction, or importance of the words. Other well-known feature extraction techniques are n-gram and PoS tagging. N-gram is the contiguous sequence of n items in a text. It identifies the neighboring sequences of items in a document [Singh et al. 2019], [Suciati and Budi 2020], [Singh et al. 2021]. PoS tagging labels the words into speech categories such as nouns, verbs, articles, adjectives, etc. [Gundapu and Mamidi 2020], [Birjali et al. 2021]. In some studies, opinion words, word count, and negation terms were used as features [Pravalika et al. 2017], [Bohra et al. 2018].

The extracted features can be irrelevant and redundant; hence, features 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. Feature selection techniques can be classified into Filter and Wrapper methods. The Filter method is comparatively fast since it identifies the best features by their correlation with the dependent variable. In contrast, the wrapper method trains a model to identify the relevant and useful features [Singh et al. 2019], [Gundapu and Mamidi 2020], [Birjali et al. 2021].

The last step is sentiment classification which identifies opinions and classifies them as positive, negative, neutral, hate, good, bad, etc. [D'Andrea et al. 2015]. Machine Learning-based or Lexicon-based approaches can be used to determine the opinion. Machine Learning-based approaches train and test the data set to identify the polarity, while Lexicon-based approaches use dictionaries. Section 4 discusses these two approaches in more detail.

Once the sentiment classification is finished, the results can be evaluated by using indexes such as Precision, Recall, Accuracy, and F1-Score [Kharde and Sonawane 2016], [Singh et al. 2019], [Gundapu and Mamidi 2020].

4 Levels of Sentiment Analysis

According to the task, there are three levels of SA, document level, sentence level, and aspect level [Birjali et al. 2021].

Document level SA considers the whole document as a basic information unit and identifies the sentiment. For example, document level SA determines the overall sentiment in a product or service review. This level is best for documents written by a single person and unsuitable for documents that compare multiple entities or contain opposite sentiments [Birjali et al. 2021]. The authors [Kumar et al. 2018] implemented an aggression annotated data set for Hindi-English code-mixed text. The annotation was done at the document level, where a complete post, comment, or discourse unit was considered a document.

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 determining the sentiment of a subjective sentence as positive, negative, or neutral [Ahmad et al. 2019], [Birjali et al. 2021]. The study [Shalini et al. 2018] presented an 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 authors [Kazi et al. 2020] conducted a sentence level language identification for the Gujarati-Hindi code-mixed text. The study used seven classes to label the sentences according to the language.

Although SA at the previous two levels is important, 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 movie aspects, 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 [Joshi et al. 2017], [Ahmad et al. 2019], [Birjali et al. 2021]. The study [Suciati and Budi 2020] proposed an aspect-based SA and emotion detection approach for restaurant reviews written in Indonesian-English code-mixed text. The study considered different aspects, such as food, price, service, and ambiance. The data set consisted of 14103 reviews, tagged with both sentiment and emotion labels. According to the label distribution, it was noticed that the data set was imbalanced in both labels. All the aspects except food were dominated by ‘neutral’ in both sentiment and emotion. The positive sentiment and happy emotion dominated the food aspect. The study [Arianto and Budi 2020] presented an aspect-based SA on Google Maps reviews written in Indonesian-English code-mixed text. The aspects used were attractions, amenities, accessibility, image, price, and human resources. Three sentiment polarities were used in the study, and each review was annotated with the relevant polarities of all six features. Annotated data of the image aspect had different annotation results between annotators which led to the removal of some data from the study. This made the Machine Learning models perform poorly on the image aspect. Hence, compared to other aspects, the highest score achieved by the image aspect was low.

5 Approaches of Sentiment Analysis

Literature divides SA approaches into two categories: Machine Learning-based and Lexicon-based approaches. Figure 1 shows the outline of SA approaches [Mahadzir et al. 2021].

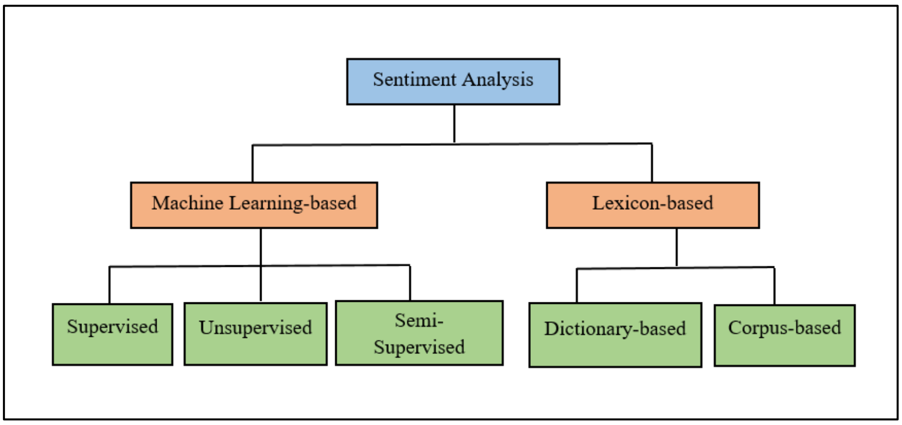

Figure 1: Approaches of Sentiment Analysis

5.1 Machine Learning-Based Approach

Machine Learning-based approaches train and test data sets to identify the sentiment polarity. It can locate domain-specific patterns and create models for specific contexts [Birjali et al. 2021]. Machine Learning-based approaches work well with multilingual data (data consisting of multiple languages) as the data sets can be trained using Machine Learning algorithms to classify the sentiments. The success of Machine Learning approaches relies on the quality and the quantity of the data set [Konate and Du 2018], [Sharounthan et al. 2021], [Srinivasan and Subalalitha 2023]. The main drawback of this approach is that a trained classifier only works well with the particular data set; hence, the same classifier cannot adapt to a new data set or domain [Birjali et al. 2021]. This approach can be divided into three parts: Supervised, Unsupervised, and Semi-supervised Learning [Mahadzir et al. 2021].

5.1.1 Supervised Learning

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, neutral, etc. [Mahadzir et al. 2021]. The most commonly used supervised learning classifiers are SVM, Artificial Neural Network (ANN), NB, Bayesian Network (BN), and Maximum Entropy (ME) [Birjali et al. 2021].

Authors [Singh et al. 2019] analyzed the sentiment of agriculture-related comments written in English-Punjabi code-mixed text. The extracted comments were classified sentence-wise as positive, negative, and neutral. Features such as the number of words that match English-Punjabi sentiment words, the number of ill words, the number of character repetitions, and n-grams were used. SVM and NB algorithms were used to train the model. The research initially tested the pipeline using unigram and later by n-grams. The study identified that the performance was enhanced with the n-gram model.

5.1.2 Unsupervised Learning

Since supervised learning needs a labeled corpus for training, the data need to be collected and annotated. This is a challenging, time-consuming, and labor-intensive process, especially when the text is unstructured. Unsupervised learning can be used in such situations since it does not need prior training with a labeled corpus. Unsupervised learning algorithms can find hidden patterns in a given data set without guidance. Usually, unsupervised learning uses statistical approaches or clustering algorithms [Joshi et al. 2017], [Birjali et al. 2021].

An unsupervised SA was implemented [Yadav and Chakraborty 2020] for the Spanish-English code-mixed text. The study used multilingual and cross-lingual embeddings and analyzed the code-mixed text in a zero-shot way. The study achieved an F1-score of 0.58 without parallel corpus and 0.62 with parallel corpus on the same benchmark in a zero-shot way. The study showed that zero-shot approaches can transfer knowledge from monolingual text to code-mixed text using embeddings and models without losing performance.

5.1.3 Semi-Supervised Learning

Semi-supervised learning is used in similar situations to unsupervised learning, where acquiring a labeled data set is challenging. However, this method differs from unsupervised learning as it initially needs a small labeled data set for training. Hence, the technique 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 using more unlabelled data and can even create more generalized classifiers [Ahmad et al. 2019], [Birjali et al. 2021].

The study [Lo et al. 2016] implemented a multilingual semi-supervised approach to detect the polarity in Singaporean English (Singlish) text. For constructing an annotated data set, the study 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 were used. The study 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. The study encountered challenges with ambiguous words such as sarcastic expressions and localized named entities. The authors suggested that disambiguation techniques improve accuracy, especially on code-mixed text with positive polarity.

5.2 Lexicon-Based Approach

Instead of training, a lexicon is used to identify the polarity values in the Lexicon-based approach. A 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 [Birjali et al. 2021]. This approach is more suitable for monolingual data (data consisting of a single language) as the standard lexicons are available. Two techniques of the Lexicon-based approach are the Dictionary-based approach and the Corpus-based approach. The dictionary-based approach assumes that synonymous have equal sentiments and antonyms have opposite sentiments. The first step of the approach starts by manually collecting the initial seed words with their known sentiments. The list grows by searching the synonymous and antonyms of the previous list over the lexicons, and the newly found words are added to the initial list. This iteration continues until no new words are found [Joshi et al. 2017], [Birjali et al. 2021]. The dictionary-based approach can quickly build a lexicon with a large number of words and sentiment polarities. The corpus-based approach starts with a list of seed words and uses the syntactic or co-occurrence patterns to search for the other sentiment words in a large corpus [Mahadzir et al. 2021].

One of the problems associated with the Lexicon-based 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" expresses a positive sentiment, while in the second sentence, the word conveys a negative sentiment. Hence, a word can have different senses according to the domain; thus, a positive word in a specific domain can be a negative word in another domain. This challenge can be handled using a domain-specific sentiment lexicon [Joshi et al. 2017], [Birjali et al. 2021]. 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 [Birjali et al. 2021].

The paper [Singh et al. 2021] used a statistical technique to perform an SA on agriculture-related comments written in English-Punjabi code-mixed text. The study created a dictionary of English-Punjabi code-mixed text and categorized the words into positive, negative, and neutral sentiments by assigning the polarity values ranging from [-1, 1]. A statistical technique was used on the dictionary-based data set at the sentence level and achieved the highest Accuracy of 83% with the trigrams approach. The research [Tho et al. 2021] presented an SA for the Indonesian and Javanese code-mixed text using a Lexicon-based approach. The study 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. However, both lexicons did not perform well with positive and neutral sentiments. The reason is many Indonesian and Javanese code-mixed text consists of words with positive sentiments which cause false positive results.

6 Challenges of Sentiment Analysis

Due to the complexities associated with languages, SA deals with different challenges. This section highlights the key challenges of SA that make it difficult to analyze the text and detect the sentiment polarities.

6.1 Sarcasm Detection

Sarcasm refers to saying or writing something that means the opposite of what it seems to say. The difficulty and ambiguity of sarcasm make SA a very challenging task. 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 the sentence expresses a negative sentiment. For this reason, it should identify the meaning in these types of sentences rather than detect the syntaxes [Kharde and Sonawane 2016], [Joshi et al. 2017], [Birjali et al. 2021].

The study [Aggarwal et al. 2020] proposed a Deep Learning approach to detect sarcasm in Hindi-English code-mixed text. The authors used two-word embedding approaches, Word2Vec and FastText. They experimented with different Deep Learning models: CNN, Long Short-Term Memory (LSTM), and Bidirectional LSTM (BiLSTM) (with and without attention), and achieved the highest Accuracy of 78.49% with attention-based BiLSTM. The authors [Swami et al. 2018] created the first English-Hindi code-mixed data set for sarcasm detection and 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. However, the data set contained 504 sarcastic tweets only.

6.2 Negation Handling

Words such as "not", "no", "never", and "cannot" are some common examples of negative words. Negative words reverse the sentiment polarities and change the opinion orientation. For example, "She is a good girl" expresses a positive sentiment, while "She is not a good girl" expresses a negative opinion. In some cases, negative words are contained in the stop-word list and hence removed during the preprocessing step or implicitly ignored as they have a neutral polarity in the lexicon. Negations cannot be simply handled by reversing the sentence's polarity since sometimes they may not affect the overall polarity of the sentence [Joshi et al. 2017], [Birjali et al. 2021].

The authors [Bohra et al. 2018] used negative words as a feature in hate speech detection on Hindi-English code-mixed text. The study counted the number of negative words in a tweet and considered it as a feature. The study used SVM and RF classifiers, and the negation feature achieved a similar Accuracy of 63.6 for both classifiers.

6.3 Ambiguity

Ambiguity can be divided into two parts as Structural Ambiguity (Syntactic Ambiguity) and Lexical Ambiguity (Semantic Ambiguity) [Arukgoda et al. 2014].

Structural ambiguity results from different meanings of a sentence [Arukgoda et al. 2014]. Even though the sequence of words is similar, the sentence is interpreted differently as the sentence may have different syntactic structures in different situations. For example, "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) [Arukgoda et al. 2014].

The paper [Smith and Thayasivam 2019] 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 English words, such as "shape", and "royal" have multiple meanings when used in Sinhala-English code-mixing. In addition to that, Sri Lankans usually use "k" to represent the English word "okay". However, people use "k" at the end of numbers such as "100k" where the value expressed is 100, not 100000. The authors noticed that ambiguous words make SA of 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 was implemented [Aggarwal et al. 2018] using a four-step process. In the first two steps, the language of each word was recognized, and candidate pun locations were identified. In the third step, the left and right contexts of the candidate pun locations were looked up, and all possible words that may occur at the location were identified. Finally, 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. However, the model failed when a word in the pun was translated to multiple words in the target language. The model was also unable to work when a pun was based on the pronunciation of an abbreviation. Further, the model failed when a phrase is unusual.

6.4 Low-Resource Languages

SA is an almost solved problem for a language like English, for which a large number of linguistic resources are available. However, linguistic resources are scarce for languages like Sinhala and Bambara [Konate and Du 2018], [Smith and Thayasivam 2019]. Most SA studies are based on supervised learning approaches that rely highly on linguistic resources. Therefore, applying supervised learning approaches to low-resource languages is extremely costly. However, using unsupervised or semi-supervised approaches or constructing linguistic resources from scratch would help to overcome the challenge [Joshi et al. 2017], [Birjali et al. 2021].

The study [Konate and Du 2018] implemented an SA on code-mixed Bambara-French text. They 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 instead of pre-trained word vectors. The study achieved the highest Accuracy of 83.23% with the one-layer CNN Deep Learning model. The research used an imbalanced data set that contained fewer negative comments compared to positive and neutral comments. The authors [Smith and Thayasivam 2019] presented the first language detection model to detect Sinhala-English code-mixed text. Since this was a novel approach, the data set was newly built by scrapping Facebook chats and posts. Manual annotation was done in two phases, annotated sentences to identify the code-mixed text and annotated each word of code-mixed text with language tags. The study developed an XGBoost (XGB) model with 92.1% Accuracy and a CRF model with an F1-score of 0.94 for sequence labeling. The authors recognized that tree-based models are more suitable for code-mixed text classification compared to Machine Learning models. However, the data set used was insufficient to train the tree-based models to perform well with sequence tagging.

6.5 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 context. For example, “fast” is recognized as a negative word in the teaching domain, but it is expressed as a positive sentiment in the phone domain [Sharounthan et al. 2021]. Therefore, the domain or context must be considered when the sentiment polarity is calculated [Kharde and Sonawane 2016], [Joshi et al. 2017].

The study [Pravalika et al. 2017] presented a domain-specific SA for the Hindi-English code-mixed text. They proposed a hybrid system that incorporates Lexicon-based and Machine Learning-based approaches. In the Lexicon-based approach, a lexicon representing the movie domain was 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 gained 72%.

7 Comparison of Research Findings

According to the literature found, it was identified that code-mixed text-related studies were mainly focused on four areas: (a). Preprocessing, (b). Language identification, (c). Corpus creation, (d). Sentiment or Emotion classification [Mahadzir et al. 2021].

7.1 Preprocessing

The studies on preprocessing were mainly focused on tasks such as noisy text identification, spell correction, and stop word removal [Mahadzir et al. 2021]. The authors [Dutta et al. 2015] tried to correct the misspelled English words in Bangla-English code-mixed text through word level language identification. The identified English words that did not appear in the vocabulary were considered misspelled and directed to a spell checker. The spell checker was based on the noisy channel model and tackled wordplay, contracted words, and phonetic variations. The spell checker obtained the Accuracy of 69.43%. However, the checker was unable to handle the words with more than two errors and words with punctuation marks. The study [Barik et al. 2019] proposed a pipeline to normalize the Indonesian-English code-mixed tweets using four modules, tokenization, language identification, lexical normalization, and translation. The tweets were tokenized in the first two modules, 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 the Indonesian language. The pipeline achieved a score of 54.07 for Bilingual Evaluation Understudy (BLEU) and a score of 31.89 for Word Error Rate (WER). The overall performance of the pipeline was low since the final result depended on the output of previous modules. The error of each module propagated to the next modules.

7.2 Language Identification

Language identification is considered a challenging task in social media code-mixed contexts. The study [Mandal et al. 2018a] presented a word level language identification for the Bengali-English code-mixed text. They built two LSTM models using character and phonetic encoding, combined them, and implemented two ensemble models using the stack and threshold techniques. The stacking model achieved an Accuracy of 91.78%, and the threshold model achieved an Accuracy of 92.35%. The study was unable to capture the context information since the experiments were conducted on the word level instead of the sentence level. Further, the system was incapable of handling elongated words and words with numeric or special characters. The study [Shanmugalingam and Sumathipala 2019] 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. However, the RF classifier didn't categorize the "rest" tags such as named entities, acronyms, and other language tags accurately. Authors [Gundapu and Mamidi 2020] experimented with different models for language identification in English-Telugu code-mixed data. The CRF model gave the best output with an F1-score of 0.91. The research encountered problems with the romanization of Telugu words and different social media syntaxes. The study used a comparatively smaller corpus that contained 1987 code-mixed sentences. The study [Kazi et al. 2020] conducted a language identification for the Gujarati-Hindi code-mixed text. The languages were identified at the sentence level by using seven classes. The study used six Machine Learning classifiers and achieved the highest accuracy of 92% with SVM. Decision Tree (DT) performed worst on the data set. The research used a highly imbalanced and unstructured corpus which affected the performance of the predicted models.

7.3 Corpus Creation

Although various corpora are available for monolingual languages such as English, Russian, Norwegian, Hindi, etc., a limited number of corpora and lexicon resources are available for code-mixed language pairs. Therefore, corpus creation is one of the significant tasks in code-mixed-based studies. The study [Chakravarthi et al. 2020b] created an annotated Tamil-English code-mixed corpus with 15,744 comments. The comments were collected using the YouTube comment scraper tool and filtered out the non-code-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. As a benchmark, the study applied some Machine Learning algorithms on the corpus to determine the sentiments. All the classification algorithms performed poorly on the data set. The authors suggested that the class imbalance problem caused this poor performance. The authors [Mandal et al. 2018b] prepared a Bengali-English code-mixed corpus using two phases of annotation: 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 language tagger achieved the Accuracy of 81%, and the sentiment tagger achieved the Accuracy of 80.97%. The authors [Bohra et al. 2018] created a corpus and identified hate speeches in Hindi-English code-mixed text using 4574 tweets. The annotation was 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. Authors [Chakravarthi et al. 2022] presented an annotated data set for three language pairs, namely, Tamil-English, Kannada-English, and Malayalam-English. The data set contained around 60000 code-mixed comments annotated for both SA and offensive language identification. However, the corpus was imbalanced for all three language pairs. It contained more positive samples compared to any other class in all language pairs. The study [Chakravarthi et al. 2020a] created a code-mixed corpus for the Malayalam-English, with an inter-annotator agreement of 0.8. The corpus contained 6738 comments on movie trailer reviews. In the annotation process, they identified some ambiguity issues, such as commentators comparing a movie with other movies and commenting on the different aspects of films in the same sentence. These issues made it challenging to identify the actual sentiment expressed by the viewer.

7.4 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, sad, etc. The paper [Sreelakshmi et al. 2020] implemented a model to detect hate speeches in Hindi-English code-mixed text. The study used Facebook's pre-trained library fastText to identify hate speeches. The proposed model was compared with word2vec and doc2vec algorithms and identified that the performance of the implemented model is high. They also observed that character level features give more details than word and document level features in the code-mixed classification. The study [Sasidhar et al. 2020] detected emotions for the Hindi-English code-mixed text by creating a corpus with 12000 comments. They used three classes and maintained an equal number of comments for each class to omit the class imbalance problem. A bilingual pre-trained model was retrained using Word2Vec to convert texts into vectors. Different Deep Learning models were used, and CNN-BiLSTM achieved the highest Accuracy of 83.21%. The study [Suciati and Budi 2020] presented an aspect-based SA and emotion detection approach for Indonesian-English code-mixed text. They created two scenarios. The transformation methods were used for multi-label classification with unigrams in the first scenario. The second scenario used Deep Learning algorithms with word embeddings. The RF achieved the highest F1-score of 88.4% with the Classifier Chain (CC) method for the food aspect and 89.54% with the Label Powerset (LP) method for the price aspect. In the service and ambiance 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 BiLSTM achieved the same F1-score of 88.16% for the food aspect. GRU performed well for the price aspect with an F1-score of 83.01%, and BiLSTM gained the highest F1-score of 89.03% and 84.78% for the service and ambiance aspects, respectively. The study [Shanmugavadivel et al. 2022] presented an SA of Tamil-English code-mixed text. The study used positive, negative, mixed feelings, and unknown state classes and implemented four approaches, Machine Learning, Deep Learning, Transfer Learning, and hybrid Deep Learning. They identified that CNN+ BiLSTM performed better with an Accuracy of 0.66.

8 Discussion

According to the findings, it was identified that feature extraction is the most critical step in the SA. Feature extraction directly impacts the performance of sentiment classification since it extracts valuable information about the characteristics of the text. Among the three levels of SA, aspect level SA is more challenging and interesting as it has to identify the sentiments of each aspect of entities. SA studies for code-mixed text were mainly centered on preprocessing, language identification, corpus creation, and sentiment classification tasks. Besides, experiments can be carried out in areas such as Named Entity Recognition (NER), code-mixed machine translation, question answering, and question classification. It was identified that Machine Learning-based approaches are more suitable for multilingual or code-mixed text, as code-mixed text often does not have standard lexicons. However, the performance of Machine Learning approaches depends on the quality and the quantity of the data set. The literature showed that from the traditional Machine Learning classifiers, RF, followed by SVM and CRF, achieved the highest Accuracy and F1-scores. From Neural Network approaches, BiLSTM (BLSTM), followed by CNN performed well. When the data set was large, Neural Network approaches performed better than traditional Machine Learning approaches. The study encountered five significant challenges faced by SA of code-mixed text: sarcasm detection, negation handling, ambiguity, lack of linguistic resources, and domain dependency. These challenges demonstrate that SA of code-mixed text remains an open research field. Among the challenges presented, the major challenge was identified as sarcasm detection since models may incorrectly classify the sentiment of sarcastic sentences. Due to the lack of linguistic resources available, the studies based on SA of code-mixed text are still at the beginning for some language pairs such as Bambara-French, Indonesian-Javanese, Sinhala-English, etc. However, a vast interest can be observed in Spanish-English, and Indian language pairs such as Hindi-English, and Bengali-English. It was observed that most of the studies related to the SA of code-mixed text faced difficulties with smaller data sets and imbalanced data sets; hence, some tags were misclassified and achieved low performances.

9 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 tremendous attention. This paper attempted to study the literature on the state-of-the-art in SA of code-mixed text. This review included 29 primary studies published from 2015 to 2022. The generic process, levels, basic approaches, and challenges of SA of code-mixed text were highlighted in the paper. The paper also discussed, summarized, and compared language pairs, tasks, approaches, performances, and limitations of various code-mixed-based studies.

Spanish-English, Hindi-English, and Bengali-English are the most tackled language pairs in the field of SA of code-mixed text. But recently, other language pairs have also gained more attention. Linguistic resources for these language pairs are still scarce. Hence, addressing those other language pairs by creating linguistic resources such as balanced datasets and lexicons can be conducted as future research.

References

[Aggarwal et al. 2018] Aggarwal, S., Mathur, K., Mamidi, R.: "Automatic Target Recovery for Hindi-English Code Mixed Puns"; arXiv preprint arXiv:1806.04535 (2018).

[Aggarwal et al. 2020] Aggarwal, A., Wadhawan, A., Chaudhary, A., Maurya, K.: "“Did You Really Mean What You Said?” : Sarcasm Detection in Hindi-English Code-Mixed Data Using Bilingual Word Embeddings"; arXiv preprint arXiv:2010.00310 (2020).

[Ahmad et al. 2019] Ahmad, G.I., Singla, J., Nikita, N.: "Review on Sentiment Analysis of Indian Languages with a Special Focus on Code Mixed Indian Languages"; 2019 International Conference on Automation, Computational and Technology Management (ICACTM) (2019), 352–356.

[Ahmad et al. 2022] Ahmad, G.I., Singla, J., Anis, A., Reshi, A.A., Salameh, A.A.: "Machine Learning Techniques for Sentiment Analysis of Code-Mixed and Switched Indian Social Media Text Corpus: A Comprehensive Review"; International Journal of Advanced Computer Science and Applications, 13, 2 (2022).

[Arianto and Budi 2020] Arianto, D., Budi, I.: "Aspect-Based Sentiment Analysis on Indonesia's Tourism Destinations Based on Google Maps User Code-Mixed Reviews (Study Case: Borobudur and Prambanan Temples)"; Proceedings of the 34th Pacific Asia Conference on Language, Information and Computation (2020), 359–367.

[Arukgoda et al. 2014] Arukgoda, J., Bandara, V., Bashani, S., Gamage, V., Wimalasuriya, D.: "A Word Sense Disambiguation Technique for Sinhala"; 2014 4th International Conference on Artificial Intelligence with Applications in Engineering and Technology (2014), 207–211.

[Barik et al. 2019] Barik, A.M., Mahendra, R., Adriani, M.: "Normalization of Indonesian-English Code-Mixed Twitter Data"; Proceedings of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019) (2019), 417–424.

[Birjali et al. 2021] Birjali, M., Kasri, M., Beni-Hssane, A.: "A Comprehensive Survey on Sentiment Analysis: Approaches, Challenges and Trends"; Knowledge-Based Systems, 226 (2021), 107134.

[Bohra et al. 2018] Bohra, A., Vijay, D., Singh, V., Akhtar, S.S., Shrivastava, M.: "A Dataset of Hindi-English Code-Mixed Social Media Text for Hate Speech Detection"; Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (2018), 36–41.

[Chakravarthi et al. 2020a] Chakravarthi, B.R., Jose, N., Suryawanshi, S., Sherly, E., McCrae, J.P.: "A Sentiment Analysis Dataset for Code-Mixed Malayalam-English"; arXiv preprint arXiv:2006.00210 (2020).

[Chakravarthi et al. 2020b] Chakravarthi, B.R., Muralidaran, V., Priyadharshini, R., McCrae, J.P.: "Corpus Creation for Sentiment Analysis in Code-Mixed Tamil-English Text"; Proceedings of the 1st Joint SLTU and CCURL Workshop (SLTU-CCURL 2020) (2020), 202-210.

[Chakravarthi et al. 2022] Chakravarthi, B.R., Priyadharshini, R., Muralidaran, V., Jose, N., Suryawanshi, S., Sherly, E., McCrae, J.P.: "DravidianCodeMix: Sentiment Analysis and Offensive Language Identification Dataset for Dravidian Languages in Code-Mixed Text"; Language Resources and Evaluation, 56, 3 (2022), 765-806.

[D'Andrea et al. 2015] D'Andrea, A., Ferri, F., Grifoni, P., Guzzo, T.: "Approaches, Tools and Applications for Sentiment Analysis Implementation"; International Journal of Computer Applications, 125, 3 (2015).

[Dutta et al. 2015] Dutta, S., Saha, T., Banerjee, S., Naskar, S.K.: "Text Normalization in Code-Mixed Social Media Text"; 2015 IEEE 2nd International Conference on Recent Trends in Information Systems (ReTIS) (2015), 378–382.

[Gundapu and Mamidi 2020] Gundapu, S., Mamidi, R.: "Word Level Language Identification in English Telugu Code Mixed Data"; arXiv preprint arXiv:2010.04482 (2020).

[Hidayatullah et al. 2022] Hidayatullah, A.F., Qazi, A., Lai, D.T.C., Apong, R.A.: "A Systematic Review on Language Identification of Code-Mixed Text: Techniques, Data Availability, Challenges, and Framework Development"; IEEE Access (2022).

[Joshi et al. 2017] Joshi, M., Prajapati, P., Shaikh, A., Vala, V.: "A Survey on Sentiment Analysis"; International Journal of Computer Applications, 163, 6 (2017), 34–38.

[Kazi et al. 2020] Kazi, M., Mehta, H., Bharti, S.: "Sentence Level Language Identification in Gujarati-Hindi Code-Mixed Scripts"; 2020 IEEE International Symposium on Sustainable Energy, Signal Processing and Cyber Security (iSSSC) (2020), 1–6.

[Keele 2007] Keele, S.: "Guidelines for Performing Systematic Literature Reviews in Software Engineering"; (2007).

[Kharde and Sonawane 2016] Kharde, V.A., Sonawane, S.S.: "Sentiment Analysis of Twitter Data: A Survey of Techniques"; International Journal of Computer Applications, 139, 11 (2016).

[Konate and Du 2018] Konate, A., Du, R.: "Sentiment Analysis of Code-Mixed Bambara-French Social Media Text Using Deep Learning Techniques"; Wuhan University Journal of Natural Sciences, 23, 3 (2018), 237–243.

[Kovács et al. 2021] Kovács, G., Alonso, P., Saini, R.: "Challenges of Hate Speech Detection in Social Media"; SN Computer Science, 2 (2021), 1-15.

[Kumar et al. 2018] Kumar, R., Reganti, A.N., Bhatia, A., Maheshwari, T.: "Aggression-Annotated Corpus of Hindi-English Code-Mixed Data"; arXiv preprint arXiv:1803.09402 (2018).

[Lo et al. 2016] Lo, S.L., Cambria, E., Chiong, R., Cornforth, D.: "A Multilingual Semi-Supervised Approach in Deriving Singlish Sentic Patterns for Polarity Detection"; Knowledge-Based Systems, 105 (2016), 236–247.

[Mahadzir et al. 2021] Mahadzir, N.H., Omar, M.F., Nawi, M.N.M., Salameh, A.A., Hussin, K.C.: "Sentiment Analysis of Code-Mixed Text: A Review"; Turkish Journal of Computer and Mathematics Education, 12, 3 (2021), 2469–2478.

[Mandal et al. 2018a] Mandal, S., Das, S.D., Das, D.: "Language Identification of Bengali-English Code-Mixed Data Using Character & Phonetic Based LSTM Models"; arXiv preprint arXiv:1803.03859 (2018).

[Mandal et al. 2018b] Mandal, S., Mahata, S.K., Das, D.: "Preparing Bengali-English Code-Mixed Corpus for Sentiment Analysis of Indian Languages"; arXiv preprint arXiv:1803.04000 (2018).

[Mishra et al. 2018] Mishra, P., Danda, P., Dhakras, P.: "Code-Mixed Sentiment Analysis Using Machine Learning and Neural Network Approaches"; arXiv preprint arXiv:1808.03299 (2018).

[Pravalika et al. 2017] Pravalika, A., Oza, V., Meghana, N.P., Kamath, S.S.: "Domain-Specific Sentiment Analysis Approaches for Code-Mixed Social Network Data"; 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (2017), 1–6.

[Qazi et al. 2017] Qazi, A., Raj, R.G., Hardaker, G., Standing, C.: "A Systematic Literature Review on Opinion Types and Sentiment Analysis Techniques: Tasks and Challenges"; Internet Research (2017).

[Sasidhar et al. 2020] Sasidhar, T.T., Premjith, B., Soman, K.P.: "Emotion Detection in Hinglish(Hindi+English) Code-Mixed Social Media Text"; Procedia Computer Science, 171 (2020), 1346–1352.

[Shalini et al. 2018] Shalini, K., Ravikurnar, A., Vineetha, R.C., Reddy, A., Kumar, A., Soman, K.P.: "Sentiment Analysis of Indian Languages Using Convolutional Neural Networks"; 2018 International Conference on Computer Communication and Informatics (ICCCI) (2018), 1–4.

[Shanmugalingam and Sumathipala 2019] Shanmugalingam, K., Sumathipala, S.: "Language Identification at Word Level in Sinhala-English Code-Mixed Social Media Text"; 2019 International Research Conference on Smart Computing and Systems Engineering (SCSE) (2019), 113–118.

[Shanmugavadivel et al. 2022] Shanmugavadivel, K., Sampath, S.H., Nandhakumar, P., Mahalingam, P., Subramanian, M., Kumaresan, P.K., Priyadharshini, R.: "An Analysis of Machine Learning Models for Sentiment Analysis of Tamil Code-Mixed Data"; Computer Speech & Language, 76 (2022), 101407.

[Sharounthan et al. 2021] Sharounthan, B., Nawinna, D.P., De Silva, R.: "Singlish Sentiment Analysis Based Rating for Public Transportation"; 2021 International Conference on Computer Communication and Informatics (ICCCI) (2021), 1–7.

[Singh et al. 2019] Singh, M., Goyal, V., Raj, S.: "Sentiment Analysis of English-Punjabi Code Mixed Social Media Content for Agriculture Domain"; 2019 4th International Conference on Information Systems and Computer Networks (ISCON) (2019), 352–357.

[Singh et al. 2021] Singh, M., Goyal, V., Raj, S.: "Sentiment Analysis of Social Media Tweets on Farmer Bills 2020"; Journal of Scientific Research, 65, 3 (2021), 156-162.

[Smith and Thayasivam 2019] Smith, I., Thayasivam, U.: "Language Detection in Sinhala-English Code-Mixed Data"; 2019 International Conference on Asian Language Processing (IALP) (2019), 228–233.

[Sreelakshmi et al. 2020] Sreelakshmi, K., Premjith, B., Soman, K.P.: "Detection of Hate Speech Text in Hindi-English Code-Mixed Data"; Procedia Computer Science, 171 (2020), 737–744.

[Srinivasan and Subalalitha 2023] Srinivasan, R., Subalalitha, C.N.: "Sentimental Analysis from Imbalanced Code-Mixed Data Using Machine Learning Approaches"; Distributed and Parallel Databases, 41 (2023), 37-52.

[Suciati and Budi 2020] Suciati, A., Budi, I.: "Aspect-Based Sentiment Analysis and Emotion Detection for Code-Mixed Review"; International Journal of Advanced Computer Science and Applications, 11, 9 (2020).

[Swami et al. 2018] Swami, S., Khandelwal, A., Singh, V., Akhtar, S.S., Shrivastava, M.: "A Corpus of English-Hindi Code-Mixed Tweets for Sarcasm Detection"; arXiv preprint arXiv:1805.11869 (2018).

[Thara and Poornachandran 2018] Thara, S., Poornachandran, P.: "Code-Mixing: A Brief Survey"; 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) (2018), 2382-2388.

[Tho et al. 2020] Tho, C., Warnars, H.L.H.S., Soewito, B., Gaol, F.L.: "Code-Mixed Sentiment Analysis Using Machine Learning Approach – A Systematic Literature Review"; 2020 4th International Conference on Informatics and Computational Sciences (ICICoS) (2020), 1-6.

[Tho et al. 2021] Tho, C., Heryadi, Y., Lukas, L., Wibowo, A.: "Code-Mixed Sentiment Analysis of Indonesian Language and Javanese Language Using Lexicon Based Approach"; Journal of Physics: Conference Series, 1869, 1 (2021), 012084.

[Yadav and Chakraborty 2020] Yadav, S., Chakraborty, T.: "Unsupervised Sentiment Analysis for Code-Mixed Data"; arXiv preprint arXiv:2001.11384 (2020).