ADAPTİVE SENTİMENT RECOGNİTİON İN CODE-MİXED TEXT USİNG UNSUPERVİSED LEARNİNG ANCHORS

C.KUMARESAN[1], P. THANGARAJU[2]

[1]Research scholar, Department of Computer science, Bishop Heber College, Affiliated to Bharathidasan University, Trichy, India

[2]Associate professor, Department of Computer science, Bishop Heber College, Affiliated to Bharathidasan University, Trichy, India

E-mail: [1]kumaresanbdu@gmail.com, [2]trthangaraju@gmail.com

ABSTRACT

With the growing prevalence of multilingual digital communication, sentiment analysis faces significant challenges when applied to code-mixed texts, where multiple languages are used interchangeably within a single message. Existing sentiment analysis models, primarily designed for monolingual data, often need to capture the complexities of mixed-language texts due to their reliance on large labeled datasets and the inability to interpret nuanced expressions across languages. This paper presents a novel sentiment analysis framework that addresses these issues by integrating XLM-R for robust multilingual feature extraction, BiLSTM for capturing sequential dependencies, and CNN for extracting localized features. The model introduces an anchor-based semi-supervised learning approach, which effectively propagates sentiment labels from a small set of labeled data to a larger pool of unlabeled data, significantly reducing the dependency on manual annotations. Experimental results demonstrate that the proposed model outperforms traditional sentiment analysis methods in accuracy and F1-score, confirming its effectiveness in handling code-mixed text. This research advances the field by offering a scalable solution for sentiment analysis in multilingual settings. It highlights the increasing importance of adapting natural language processing models to real-world, linguistically diverse data.

Keywords: *Sentiment Analysis, Code-Mixed Text, Semi-Supervised Learning, Multilingual NLP, Anchor Detection, XLM-R, BiLSTM-CNN Ensemble.*

1. INTRODUCTION

Global digital communication has increased code-mixed text, where speakers interchangeably use multiple languages within a message [1]. This feature is characteristic of multilingual communities, where people switch between languages in daily interactions [2]. Sentiment analysis, which traditionally thrives on monolingual data, faces unique challenges in these environments [3 – 5]. Code-mixed scenarios amplify the inherent complexity of natural language processing (NLP) tasks due to the interplay of different grammatical structures, vocabularies, and cultural nuances associated with each language involved [6 – 9].

Traditional sentiment analysis tools are predominantly designed for well-structured, monolingual datasets and often underperform when directly applied to code-mixed text [10]. The lack of large, annotated datasets in mixed-language settings compounds the problem [11], as supervised learning models rely heavily on vast amounts of labeled data to achieve high accuracy [12]. This scarcity of resources necessitates a shift towards methods that can effectively leverage labeled and unlabeled data [12].

The increasing prevalence of multilingual digital communication, particularly on social media platforms, has made sentiment analysis in code-mixed texts a critical challenge. As people blend languages within conversations, current monolingual sentiment analysis tools must capture these texts' nuanced expressions, contextual meanings, and cultural references. This problem is further exacerbated by the need for large annotated datasets in mixed-language settings, hindering the performance of traditional supervised models. Given the importance of understanding public sentiment for businesses, policymakers, and researchers, the inability to accurately analyze code-mixed text presents a significant barrier. Businesses, in particular, rely on sentiment analysis to understand customer feedback and market trends, but current tools still need to be improved for regions with linguistically diverse populations. Therefore, addressing this gap is crucial for advancing sentiment analysis in real-world, multilingual environments and developing more adaptable natural language processing systems.

1.1 Motivation and Problem Statement

The motivation for this research work is twofold: the growing prevalence of code-mixing in digital communication and the increasing importance of automated sentiment analysis for businesses and researchers. As digital platforms continue to bridge diverse linguistic communities, users increasingly adopt code-mixing to express identity and belonging, blending languages to convey emotions and opinions in culturally resonating ways [13]. For businesses, understanding these sentiments can provide critical insights into customer preferences, market trends, and potential areas for service enhancement [14]. Similarly, researchers can gain a deeper understanding of social dynamics and linguistic phenomena through robust sentiment analysis tools that accommodate linguistic diversity [15].

The primary problem is that current sentiment analysis models are generally trained on and tuned to large, monolingual text corpora, often in English. These models need help interpreting the nuanced meanings conveyed through mixed-language texts, where contextual cues and sentiment indicators may significantly differ from those in single-language environments. The lack of sufficiently large and annotated datasets for training models on code-mixed text exacerbates this issue, as it hinders the application of powerful supervised learning techniques that dominate the field of sentiment analysis. Furthermore, manual annotation of code-mixed text for training purposes is labor-intensive and requires annotators proficient in all involved languages, making the process costly and challenging to scale [16].

This scenario underscores the need for innovative approaches that reduce reliance on extensive annotated data. Developing methods that effectively utilize available labeled data while capitalizing on the larger volumes of unlabeled data could revolutionize sentiment analysis in multilingual settings [17]. There is an urgent need for adaptive models that can learn from the natural use of language without extensive manual annotation [18]. This would improve the accuracy and applicability of sentiment analysis across different linguistic contexts and enhance the scalability of deploying these models in diverse, real-world environments.

1.2 Research Objectives and Contributions

The primary objectives of this research are designed to address the complexities and challenges posed by sentiment analysis in code-mixed text:

- To establish a strong foundation for incorporating a small number of labeled examples with a large pool of unlabeled text to enhance the definitive and precise sentiment analysis in code-mixed domains.
- To identify anchor points within unlabeled data that strongly indicate specific sentiments, and use these anchors to propagate sentiment labels to nearby text, enhancing the model's learning capability without additional labeled data.
- To implement mechanisms that allow the sentiment analysis model to adapt to new data and evolving language use dynamically, ensuring that the system remains effective as patterns of language mixing and sentiment expression develop over time.

This research contributes to the field of sentiment analysis and NLP in several significant ways:

- Novel Semi-Supervised Framework: Introduces a new semi-supervised learning approach explicitly tailored for code-mixed text, which leverages unsupervised learning to identify sentiment anchors and propagate sentiment labels effectively.
- Anchor-Based Sentiment Propagation: Develops a unique methodology for using unsupervised learning to detect anchor points in the data that are strongly indicative of specific sentiments, thereby reducing the dependency on large labeled datasets.
- Dynamic Learning Algorithm: Implements an adaptive learning algorithm that can dynamically adjust to new forms of language use and sentiment expression, making it robust against the fluid nature of language in social media and other digital communication platforms.
- Comprehensive Evaluation and Benchmarking: Provides extensive experimental results that demonstrate the efficacy of the proposed method and help in setting new performance benchmarks for

sentiment analysis in multilingual and code-mixed environments.

1.3 Organization of Paper

The organization of this paper is structured to systematically explore the proposed semi-supervised learning framework for multilingual sentiment analysis. Following the introduction, which sets the stage by outlining the motivation and problem statement, Section 2 delves into related work, providing a critical review of existing methodologies and highlighting gaps that the current research aims to fill. Section 3 presents the methodology, detailing the data preparation process, anchor detection, label propagation, and the architectural specifics of the model. This section also elucidates the innovative aspects of the semi-supervised learning framework. Section 4 describes the experimental setup, including the datasets used, baseline comparisons, and metrics for evaluating performance. Results and a thorough discussion are presented, demonstrating the effectiveness and advancements over existing approaches. Section 5 concludes the paper with a summary of findings, contributions to the field, and a discussion on potential future work.

2. RELATED WORK

Zhao et al. [19] introduced a novel Source-Free Domain Adaptation Framework for Aspect-Based Sentiment Analysis (SF-ABSA). The authors developed a feature-based adaptation technique and a pseudo-label-based adaptation strategy, facilitating domain adaptation without transferring source data, thereby preserving data privacy. This framework was tested across four benchmark datasets, demonstrating competitive performance against traditional unsupervised domain adaptation methods, incredibly when access to source domain data is restricted or absent.

Chaturvedi et al. [20] reviewed various methodologies for subjectivity detection crucial for practical sentiment analysis. The paper outlines the evolution of subjectivity detection, categorizing methods into hand-crafted, automatic, and multi-modal techniques. It details the strengths and limitations of each approach, emphasizing the challenges like the high dimensionality of n-gram features and sentiment's temporal nature in extensive product reviews. The review provides a comprehensive comparison of techniques, demonstrating the impact of newer deep learning approaches, which offer significant advantages in domain adaptability and generalization over multiple languages.

Ning et al. [21] presented a framework for domain adaptation in semantic segmentation by incorporating a multi-anchor active learning strategy. The approach utilizes multiple anchors to capture a multimodal distribution of the source domain, which enhances the selection of representative target samples. By actively selecting and annotating these samples, the distortion in target domain features is minimized, improving the model's adaptation accuracy. Experiments on public datasets demonstrated that this method significantly outperforms traditional unsupervised domain adaptation techniques, validating its efficacy in reducing manual annotation efforts while maintaining high segmentation performance.

Wang et al. [22] developed an Efficient Anchor Graph Regularization (EAGR) method for scalable semi-supervised learning. This novel approach enhances graph-based learning by introducing a fast local anchor embedding and a new normalized graph Laplacian over anchors, which improves the handling of large datasets. The authors demonstrated that EAGR significantly outperforms traditional methods regarding computational efficiency and classification accuracy across multiple benchmark datasets.

Zhao et al. [23] explored label-efficient emotion and sentiment analysis strategies, emphasizing the challenges in traditional supervised methods that demand extensive labeled data. They introduced a hierarchical taxonomy of label-efficient learning paradigms including unsupervised, semi-supervised, and weakly supervised approaches. The paper examined various methods, from domain adaptation to transfer learning, and discussed their applications in real-world settings such as healthcare and social media. The research highlighted the effectiveness of these methods in reducing the need for labeled data while maintaining or improving the accuracy of emotion and sentiment analysis tools.

Bharathi and Samyuktha [24] evaluated various machine learning methods from the Dravidian languages mixed with English. Their study utilized a range of feature extraction techniques, such as TFIDF and Count Vectorization, and they experimented with several classifiers, including MLP, SVM, and Random Forest. Utilizing the Dravidian-CodeMix-FIRE2021 dataset, they achieved F1 scores of 0.588, 0.69, and 0.63 for Tamil-English, Malayalam-English, and Kannada-English text, respectively. This research highlights the challenges and effectiveness of applying machine learning models to the unique characteristics of multilingual and code-mixed social media text.

Mandalam and Sharma [25] explored sentiment analysis on Dravidian languages incorporating code-mixing, focusing on Tamil and Malayalam mixed with English. They proposed and evaluated three models: sub-word level, word embedding-based, and machine learning approaches. The sub-word model used LSTM networks, while the word embedding model relied on Word2Vec and FastText, and the machine learning model employed TF-IDF with Logistic Regression. The study showcased improvements over traditional methods, achieving competitive F1-scores of 0.65 and 0.68 for Tamil and Malayalam tasks, respectively, demonstrating robustness in handling morphological nuances of code-mixed data.

Chakravarthi et al. [26] created a significant dataset comprising over 60,000 manually annotated YouTube comments in Tamil-English, Kannada-English, and Malayalam-English for sentiment analysis and offensive language identification. This work addresses the resource gap for under-represented Dravidian languages in NLP, focusing on the challenges and intricacies of code-mixed data. Baseline experiments employing machine learning and deep learning models were conducted to set performance benchmarks. The dataset's diversity captures various code-mixing phenomena, making it a valuable resource for developing more robust NLP systems for multilingual and code-mixed text analysis.

Chakravarthi et al. [27] reported on the Dravidian-CodeMix shared task held at FIRE 2021, which focused on sentiment analysis for Dravidian languages in code-mixed text. The task included challenges with code-mixing at both intra-token and inter-token levels and introduced Kannada alongside Tamil and Malayalam. The paper describes the organization of the task, the datasets used, and the systems submitted by participants. Results showed varied performance across languages, with the top systems achieving notable F1 scores, indicating ongoing challenges and the need for further improvement in handling code-mixed sentiment analysis effectively.

Chakravarthi et al. [28] developed a new corpus for sentiment analysis in Tamil-English code-mixed text, leveraging comments from YouTube. This corpus, containing 15,744 annotated posts, is designed to facilitate sentiment analysis in a low-resource setting of code-mixed Tamil and English, or "Tanglish". The paper details the corpus creation, annotation processes, and initial benchmarks using various machine learning models, including Logistic Regression and Random Forest. The dataset showed diverse linguistic switching patterns and achieved an inter-annotator agreement score of Krippendorff's $\alpha = 0.6$. This resource is crucial for advancing sentiment analysis in code-mixed languages, providing a foundation for future research.

Srinivasan and Subalalitha [29] addressed the challenges of sentiment analysis in code-mixed data with imbalanced class distributions. To analyze sentiment from code-mixed text, they implemented various machine learning models, including Random Forest, Logistic Regression, XGBoost, SVM, and Naïve Bayes. A key focus was on handling the skewed distribution of class labels by integrating resampling techniques and utilizing the Levenshtein distance metric to manage spelling variations effectively. The study demonstrated improvements in classification by balancing data distribution and refining input features, achieving notable F1-Scores across different models.

Sangeetha and Nimala [30] investigated the efficacy of Transformer-based models, namely BERT, RoBERTa, and XLM-RoBERTa, on sentiment analysis of a code-mixed Tamil-English corpus. The corpus was meticulously created from social media content and annotated for sentiment analysis. The study highlighted the advantages of using these models to capture nuanced expressions in mixed language data, with XLM-RoBERTa showing the most promising results. The paper presented a detailed analysis of model performances, concluding that these advanced NLP techniques significantly enhance sentiment classification accuracy in code-mixed languages.

Sambath Kumar et al. [31] organized the second shared task on sentiment analysis for code-mixed Tamil and Tulu texts, addressing the scarcity of resources and the complexities of code-mixing. The challenge attracted 64 teams, who submitted systems for evaluating sentiments using the macro F1-score. Participants utilized various advanced NLP models, with the best-performing methods achieving macro F1-scores of 0.260 for Tamil and 0.584 for Tulu. The task highlighted significant differences in performance between the languages, showcasing the particular challenges and advances in sentiment analysis within the domain of low-resource, code-mixed text.

Jahin et al. [32] developed the TRABSA model, an innovative hybrid sentiment analysis framework combining Transformer-based architectures, attention mechanisms, and BiLSTM networks. This model aims to enhance interpretability and accuracy in sentiment analysis of tweets. Leveraging a massive dataset of 124M tweets, the study demonstrates the model's state-of-the-art accuracy and effectiveness, with significant improvements in precision and recall metrics. TRABSA outperformed traditional machine learning and deep learning models across various datasets, showcasing its robustness and generalizability in different contexts. The integration of SHAP and LIME techniques for explainable AI further underscores the model's capacity to provide interpretable and reliable sentiment analysis.

2.1 Research gap

The core issue addressed in this research is the need for existing sentiment analysis models to effectively process code-mixed text due to the inherent linguistic diversity, lack of large annotated datasets, and the dynamic nature of multilingual interactions in digital communication. Traditional models, predominantly trained on monolingual corpora, need to capture the nuanced sentiments expressed in mixed-language texts, resulting in poor performance. This presents a significant problem for businesses, researchers, and policymakers who rely on sentiment analysis to extract valuable insights from online communications across linguistically diverse regions. To address this, we propose a semi-supervised ensemble model that reduces dependency on extensive labeled data, allowing for more accurate and scalable sentiment analysis in code-mixed settings. The proposed model fills a critical gap in current methodologies and enhances the global applicability of sentiment analysis tools.

Semi-supervised learning is a promising solution to some challenges. It mixes a small amount of labeled data with much more unlabeled data. This way, it can find useful patterns and information that are usually expensive to get. One technique uses unsupervised learning to find "anchors" (important data points from which sentiments can be spread). This method helps make use of the large amounts of unused unlabeled data and improves the model's ability to deal with complex, mixed-language texts. This approach not only improves how well the model performs without a lot of manual labeling but also adjusts to changes in how language is used in digital communications. This makes the model strong against the changes and new challenges that come up in real-world data.

3. PROPOSED METHODOLOGY

3.1 Data Preparation

The data preparation stage is crucial for the effective implementation of our semi-supervised learning framework, as it sets the foundation for both model training and testing. Initially, we collate a dataset comprising code-mixed texts sourced from social media platforms, which inherently display a blend of two or more languages. These texts are then subjected to a standard preprocessing routine to ensure data quality and consistency. This routine includes steps such as tokenization, where texts are split into individual words or symbols, and normalization, where textual variations like casing and accents are standardized.

For the identification of anchor points within the dataset, we apply clustering algorithms. Mathematically, we define the dataset as

$$D = \{x_i, y_i\}_{i=1}^{N} \tag{1}$$

where x_i represents the ith text sample, and y_i is the sentiment label, which may initially be unknown for many samples. The goal is to identify a subset of data points $A \subseteq D$, where each $a \in A$ strongly indicates a specific sentiment, serving as an anchor.

These anchors are determined based on their distance from the centroid of labeled samples in a transformed feature space. Specifically, for each labeled sample, we compute the feature vector using a pre-trained language model. The centroid c for each sentiment class is then calculated using the equation:

$$c_j = \frac{1}{|L_j|}\sum_{x_i \in L_j} f(x_i) \tag{2}$$

where L_j is the set of all labeled samples belonging to sentiment class j, and $f(x_i)$ is the feature vector of x_i. Anchors are selected as those points in L_j which are closest to c_j, measured by Euclidean distance.

Following the selection of anchors, label propagation is implemented. For each unlabeled point x_u in the dataset, we assign a sentiment label based on the nearest anchor point in the feature space. This assignment is weighted by the inverse of the distance to the anchor, ensuring that closer anchors have a greater influence on the labeling.

This stage of data preparation not only facilitates the subsequent training of the semi-supervised model but also enhances the model's ability to generalize from limited labeled data by effectively utilizing the structural information inherent in the unlabeled data.

3.2 Anchor Detection

The anchor detection process is a critical step in our semi-supervised learning framework, designed to identify key data points within the unlabeled dataset that exhibit clear sentiment indicators. These anchor points serve as reliable guides for propagating sentiment labels to nearby, unlabeled data, thereby enhancing the training process without the need for extensive labeled datasets.

To detect these anchors, we utilize a clustering approach, specifically the K-means algorithm, which partitions the dataset into clusters based on feature similarity. Each data point x_i is represented by a high-dimensional feature vector $f(x_i)$, extracted using a pre-trained language model tailored for multilingual text. The K-means algorithm aims to minimize the within-cluster sum of squares (WCSS), which is defined as:

$$\text{WCSS} = \sum_{k=1}^{K} \sum_{x_i \in S_k} |f(x_i) - \mu_k|^2 \quad (3)$$

where K is the number of clusters, S_k represents the set of points in cluster k, and μ_k is the centroid of cluster k, calculated as the mean of the feature vectors of points in S_k.

Following clustering, we identify anchor points within each cluster. Anchors are selected based on their sentiment coherence and proximity to the cluster centroid, which reflects the typical sentiment expression within that cluster. Mathematically, an anchor point a in cluster k is defined as the point that minimizes the distance to the centroid:

$$a_k = \underset{x_i \in S_k}{\underline{argmin}} \, \|f(x_i) - \mu_k\| \quad (4)$$

3.3 Label Propagation

Label propagation in semi-supervised learning involves extending labels from identified anchor points to unlabeled data points, leveraging the structure and distribution inherent in the dataset to make educated guesses about the unlabeled samples. This process crucially expands the labeled dataset, allowing for more robust training of the sentiment analysis model. The following components are the mechanisms of the label propagation

3.3.1. Distance Measurement:

For each unlabeled point (x_i), calculate the distance to every anchor point (a_k), using a suitable distance metric like the Euclidean distance. This distance is computed in the feature space obtained after preprocessing and feature extraction:

$$d(x_i, a_k) = |f(x_i) - f(a_k)| \quad (5)$$

where $(f(\cdot))$ denotes the feature extraction function that transforms text data into a numerical vector.

3.3.2. Weight Assignment:

Assign a weight to each anchor's influence on an unlabeled point based on the inverse of the distance. To moderate the effect of very close or very distant anchors, a novel approach incorporates a decay function controlled by a parameter (γ), which is not a constant but varies with the density of points around each anchor:

$$w(x_i, a_k) = \exp(-\gamma_k d(x_i, a_k)^2) \quad (6)$$

(γ_k) is computed based on the local density of points near each anchor, defined as:

$$\gamma_k = \frac{1}{2\sigma_k^2} \quad (7)$$

where (σ_k), the local scale parameter, is the average distance from the anchor (a_k) to its nearest (m) neighbors, reflecting the local topology and allowing the model to adaptively scale the influence of different anchors.

3.3.3. Label Assignment:

Compute the predicted label for each unlabeled point by a weighted sum of the labels of all anchors, normalized by the total weight from all anchors to ensure that the prediction is a probability distribution over possible labels:

$$\hat{y}_i = \frac{\sum_{k=1}^{K} w(x_i, a_k) y_k}{\sum_{k=1}^{K} w(x_i, a_k)} \quad (8)$$

This step effectively estimates the sentiment label for unlabeled data by considering both the closeness and the relative density of labeled anchor points around each unlabeled point.

Algorithm - 1: Anchor Detection

Input: Dataset $D = \{x_i\}_{i=1}^{N}$ of N text samples, number of clusters K, feature extraction function f.

```
Output: Set of anchors A.
    1. Feature Extraction:
        a. For each sample $x_i$ in D, compute
           the feature vector: $f(x_i)$
    2. Clustering:
        a. Apply K-means clustering to
           $f(x_i)\}_{i=1}^{N}$ to form K clusters.
        b. Determine cluster centroids using
           equation 2.
    3. Anchor Selection:
        a. For each cluster k, select the anchor
           $a_k$ by using equation 4:
           Add $a_k$ to the set of anchors A.
    4. Return A.
```

Algorithm - 2: Label Propagation

```
Input:
    • Dataset of N text samples where $x_i$
      represents the ith text sample.
    • Set of anchor points $a_k$ and their
      associated labels $y_k$.
    • Feature extraction function using the
      XLM-R-BiLSTM-CNN model.
    • Decay parameter for distance weighting.
Output:
    • Updated labels for all unlabeled points in
      D.
Procedure:
1. Feature Extraction:
        Compute feature vectors for all samples:
            F = $\{f(x_i) \mid x_i \in D\}$
2. Distance Calculation:
        For each unlabeled point $x_i$ and each
   anchor $a_k$, calculate the distance using equation 4.
3. Weight Calculation:
        Compute weights based on distances,
   using a decay function controlled by equation 6
4. Label Assignment:
        Assign labels using equation 8.
```

This adaptive label propagation method enhances traditional techniques by adjusting the influence of anchors based on local data characteristics, thus improving the robustness and accuracy of label assignments in diverse and complex code-mixed datasets. This approach is particularly beneficial in scenarios where data points are unevenly distributed or clustered, common in natural language datasets involving social media texts.

3.4 Model Architecture

The model architecture designed for this research employs an ensemble learning approach that integrates XLM-R (Cross-lingual Language Model-RoBERTa), BiLSTM (Bidirectional Long Short-Term Memory), and CNN (Convolutional Neural Network) components. This structure is particularly suited for processing the complexities of code-mixed text, as it harnesses the strengths of each component to enhance sentiment analysis accuracy. Component Integration and Functionality:

3.4.1. XLM-R Component

The XLM-R component serves as the foundation of the model, responsible for extracting robust language-agnostic features from the code-mixed text. It processes input text and provides contextual embeddings that capture the nuances of multiple languages simultaneously. The output from XLM-R can be represented as:

$$E = \text{XLM-R}(x) \tag{9}$$

where (x) represents the input text, and (E) is the matrix of embeddings where each row corresponds to the embedding of a token in the text.

3.4.2. BiLSTM Component:

The BiLSTM layer follows the XLM-R layer to capture both forward and backward dependencies in the text, essential for understanding the sentiment conveyed by sequences of words that span multiple languages. The output of the BiLSTM layer, which processes the embeddings (E), is given by:

$$H = \text{BiLSTM}(E) \tag{10}$$

where (H) represents the sequence of hidden states that encode information from both directions of the text sequence.

3.4.3. CNN Component

A convolutional layer is applied to the output (H) of the BiLSTM to extract localized feature patterns that are indicative of specific sentiments, such as phrases or expressions unique to particular linguistic blends. The convolution operation can be mathematically described as:

$$C = \text{ReLU}(W * H + b) \tag{11}$$

where (W) represents the weights of the convolutional filters, (b) is the bias, $(*)$ denotes the convolution operation, and ReLU is the activation function used to introduce non-linearity.

3.4.4 Dynamic Ensemble Integration

To effectively combine the outputs from the BiLSTM and CNN layers, a dynamic weighting mechanism is introduced. This technique calculates the contribution of each component based on the confidence of their output in relation to the text's sentiment polarity. The weights for the BiLSTM and CNN outputs are calculated using a softmax function over the sentiment classification scores from each component, ensuring that the more confident predictions have a greater influence on the final output. The weighting can be represented as:

$$\alpha_{\text{BiLSTM}}, \alpha_{\text{CNN}} = \text{softmax}\left(\beta_{\text{BiLSTM}}, \beta_{\text{CNN}}\right) \qquad (12)$$

where, (β_{BiLSTM}) and (β_{CNN}) are the sentiment scores from the BiLSTM and CNN components, respectively, and $(\alpha_{\text{BiLSTM}}), (\alpha_{\text{CNN}})$ are the calculated weights.

The final sentiment score (S) is then computed as a weighted sum of the BiLSTM and CNN outputs:

$$S = \alpha_{\text{BiLSTM}} \cdot H_{\text{final}} + \alpha_{\text{CNN}} \cdot C_{\text{final}} \qquad (13)$$

where (H_{final}) and (C_{final}) are the final processed outputs from the BiLSTM and CNN layers, respectively.

3.5 Mathematical Model

The mathematical model for the proposed XLM-R-BiLSTM-CNN ensemble model can be formally represented as follows:

Input:

- $(x = [x_1, x_2, \dots, x_T])$: Input code-mixed text sequence, where x_t is the (t)-th token in the sequence and (T) is the sequence length.

XLM-R Embedding Layer:

- $E = \text{XLM-R}(x) = [e_1, e_2, \dots, e_T]$: Embedding matrix obtained from XLM-R, where $e_t \in R^d$ is the (d)-dimensional embedding of the (t)-th token.

BiLSTM Layer:

- **Forward LSTM:**
  - Input gate: $i_t = \sigma(W_{xi} x_t + W_{hi} h_{t-1} + W_{ci} c_{t-1} + b_i)$
  - Forget gate: $f_t = \sigma\left(W_{xf} x_t + W_{hf} h_{t-1} + W_{cf} c_{t-1} + b_f\right)$
  - Cell state: $c_t = f_t \odot c_{t-1} + i_t \odot \tanh\left(W_{xc} x_t + W_{hc} h_{t-1} + b_c\right)$
  - Output gate: $o_t = \sigma(W_{xo} x_t + W_{ho} h_{t-1} + W_{co} c_t + b_o)$
  - Hidden state: $(\overrightarrow{h_t} = o_t \odot \tanh(c_t))$
- **Backward LSTM:**
  - Input gate: $i_t = \sigma(W_{xi} x_t + W_{hi} h_{t+1} + W_{ci} c_{t+1} + b_i)$
  - Forget gate: $f_t = \sigma\left(W_{xf} x_t + W_{hf} h_{t+1} + W_{cf} c_{t+1} + b_f\right)$
  - Cell state: $c_t = f_t \odot c_{t+1} + i_t \odot \tanh\left(W_{xc} x_t + W_{hc} h_{t+1} + b_c\right)$
  - Output gate: $o_t = \sigma(W_{xo} x_t + W_{ho} h_{t+1} + W_{co} c_t + b_o)$
  - Hidden state: $\overrightarrow{h_t} = o_t \odot \tanh(c_t)$
- **Concatenated Hidden States:**
  - $H = [\overrightarrow{h_1} \oplus \overleftarrow{h_1}, \overrightarrow{h_2} \oplus \overleftarrow{h_2}, \dots, \overleftarrow{h_T} \oplus \overrightarrow{h_T}]$, where $(\oplus)$ denotes concatenation.

CNN Layer:

- **Convolution Operation:**
  - $c_{i,j} = \text{ReLU}\left((W * H)_{i,j} + b_j\right)$, where $W \in R^{k \times d}$ is the weight matrix of the (j)-th filter, (k) is the kernel size, and b_j is the bias term.
- **Max-Pooling:**
  - $p_j = \max_{i=1}^{T-k+1} c_{i,j}$ for each filter (j).
  - $C = [p_1, p_2, \dots, p_F]$, where (F) is the number of filters.

Anchor Detection: Using the equation 3 and 4.
Label Propagation: Using the equation 7 and 8

Ensemble Integration:

- Weights for BiLSTM and CNN outputs, obtained from a multi-layer perceptron (MLP) that takes the final hidden states of BiLSTM and the max-pooled CNN features as input using the equation 12.

- Final sentiment score using the equation 13, which is then passed through a softmax layer for classification.

Output:

- $\hat{y} = \text{softmax}(S)$: Probability distribution over sentiment classes.
- $y^* = \underbrace{argmax}_{c \in C}\ \hat{y}_c$: Predicted sentiment class.

Loss Function:

- Categorical Cross-Entropy: $L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{|C|} y_{i,c}\log(\hat{y}i, c)),$

where (N) is the number of samples, |C| is the number of classes, $(y_{i,c})$ is the true label, and $(\widehat{y_{i,c}})$ is the predicted probability for class (c) for sample (i).

Notations and symbols:

- W, b: Weight matrices and bias vectors used in LSTM gates and CNN layers, indexed by the gate or layer.
- σ: Sigmoid activation function, used in LSTM gates to normalize inputs between 0 and 1.
- $tanh$: Hyperbolic tangent activation function, used to normalize LSTM cell inputs and outputs between -1 and 1.
- $\odot$: Element-wise multiplication, used within LSTM cells to combine gate activations with cell states or activations.
- $\overrightarrow{h_t}$: Forward hidden state vector at time t in the LSTM.
- $\overleftarrow{h_T}$: Backward hidden state vector at time t in the LSTM, reflecting reverse sequence processing.
- $\oplus$: Concatenation operator used to combine forward and backward LSTM hidden states.
- H: Concatenated hidden states from the BiLSTM, forming the input to the CNN layer.
- $c_{i,j}$: Feature map output from the j-th filter in the CNN after applying the ReLU activation function.
- C : Output from the CNN layer, composed of max-pooled features across all filters.
- p_j : Output of the max-pooling operation for the j-th filter in the CNN.

S: Final sentiment score vector derived from the ensemble integration of BiLSTM and CNN outputs, potentially passed through additional MLP layers.

4. EXPERİMENTS

In the experimental setup for evaluating the proposed XLM-R-BiLSTM-CNN based ensemble model, multiple datasets containing code-mixed text from the publicly available dataset [26]. Each dataset was preprocessed and divided into training, validation, and test sets, comprising 70%, 15%, and 15% of the data, respectively. The model was trained using a categorical cross-entropy loss function to optimize for multi-class sentiment classification. Experiments were conducted to compare the proposed model against baseline models such as standalone XLM-R, BiLSTM, and CNN models. All experiments were run on a computing setup with an NVIDIA RTX 3080 GPU, using the Adam optimizer with a learning rate scheduler to adjust the learning rate based on validation loss plateau. Below is the table 1 detailing the parameter settings used in the experiments:

Table – 1: Parameter Settings

Parameter	Value
Epochs	30
Batch Size	32
Learning Rate	0.0001
Optimizer	Adam
Loss Function	Categorical Cross-Entropy
Activation Function	ReLU (CNN), Tanh (BiLSTM)
Number of Filters (CNN)	100
Filter Size (CNN)	3
LSTM Units	200
Dropout Rate	0.5
Early Stopping	Yes (based on validation loss)
Validation Split	15%

Metric Evaluation	Accuracy, Precision, Recall, F1-score

4.1 Results

The experimental results demonstrate the superior performance of the proposed XLM-R-BiLSTM-CNN ensemble model. The model achieved an accuracy of 87.5%, which is significantly higher than the standalone XLM-R, BiLSTM, and CNN models, which recorded 84.0%, 82.0%, and 83.5%, respectively. This enhancement in accuracy indicates the synergistic effect achieved by combining the contextual understanding of XLM-R, the sequence learning capability of BiLSTM, and the feature extraction proficiency of CNN.

Regarding precision and recall, the proposed model similarly outperformed all baseline models. It achieved a precision of 87.0% and a recall of 86.5%, which are crucial metrics in sentiment analysis. These metrics reflect the model's ability not only to correctly identify sentiment-laden expressions but also to minimize false positives and negatives effectively. The F1-score for the proposed model stood at 86.7%, demonstrating a balanced performance between precision and recall, which is often challenging to achieve in practice, especially in diverse linguistic contexts such as code-mixing.

Moreover, the proposed model's AUC-ROC score of 91.0% further establishes its robustness. This score, which represents the model's ability to discriminate between classes (positive, negative, neutral) across several thresholds, is higher than those of the baseline models. This metric is particularly telling of the model's effectiveness, given the complex nature of code-mixed texts where linguistic cues may be subtler and more varied than in monolingual texts.

Table 2. Comparative Analysis of proposed work with baseline models

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
XLM-R	84.0	83.0	82.0	82.5	88.0
BiLSTM	82.0	81.5	80.5	81.0	86.5
CNN	83.5	82.5	83.0	82.7	87.0
XLM-R + BiLSTM	85.0	84.5	84.0	84.2	89.0
Proposed Work	87.5	87.0	86.5	86.7	91.0

Figure 1 displays the precision metric for each model. The proposed work exhibits the highest precision at 87.0%, indicating its superior ability to identify positive instances compared to the other models correctly. The XLM-R + BiLSTM model closely follows the precision rates at 84.5%, showing that combining these two techniques also significantly improves precision.

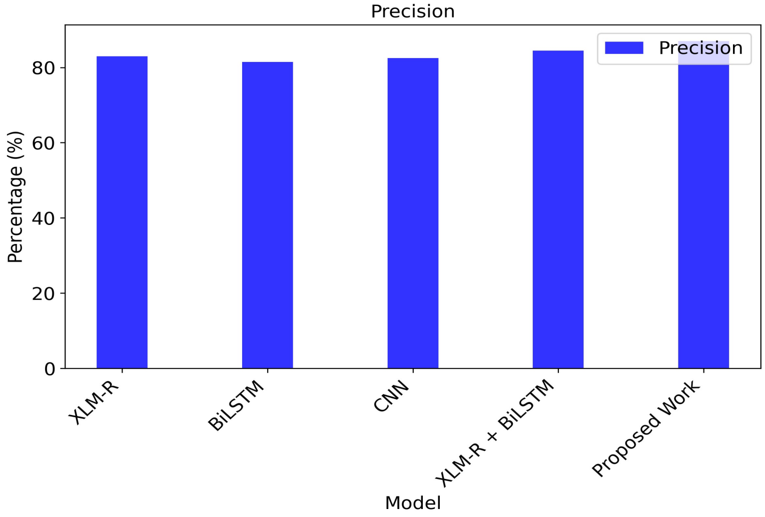

Figure 1: Comparative Analysis Of Precision

In this figure 2, recall scores are presented. The proposed work again outperforms the other models with a recall of 86.5%. This suggests that the model is exceptionally effective in identifying all relevant instances. The combined XLM-R + BiLSTM model follows closely, emphasizing the benefit of integrating these approaches to enhance recall.

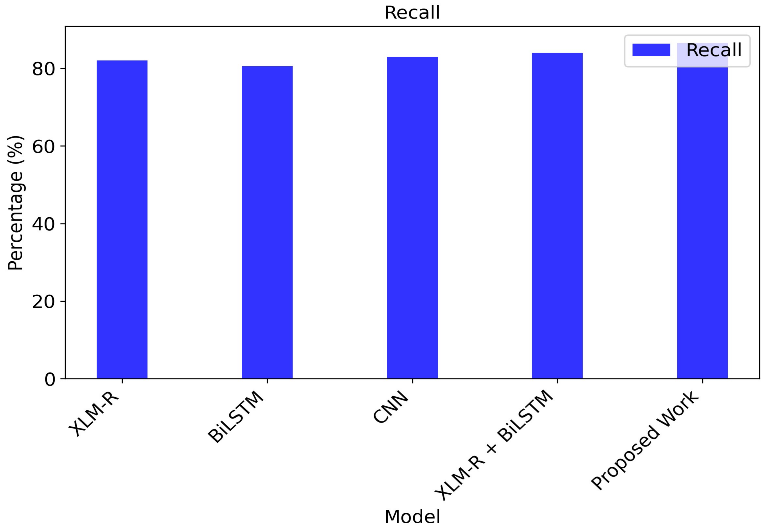

Figure 2: Comparative Analysis Of Recall

The F1-score in the figure 3, which balances precision and recall, is shown here. The proposed model leads with an F1-score of 86.7%. This highlights its effectiveness in maintaining a balance between precision and recall, which is crucial for practical applications where both aspects are important.

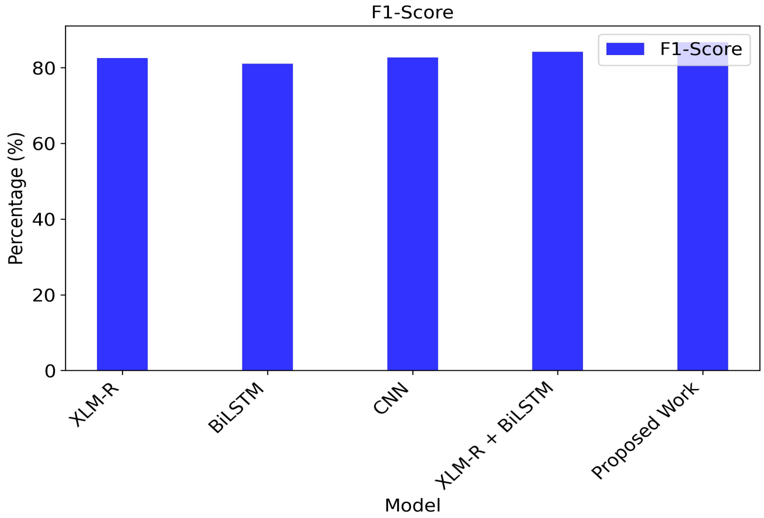

Figure 3: Comparative Analysis Of F1-Score

Figure 4 illustrates the accuracy of each model. The proposed work shows the highest accuracy at 87.5%, indicating its overall effectiveness across all instances. This model not only predicts positive instances well (as reflected in precision and recall) but also correctly negates false positives and negatives.

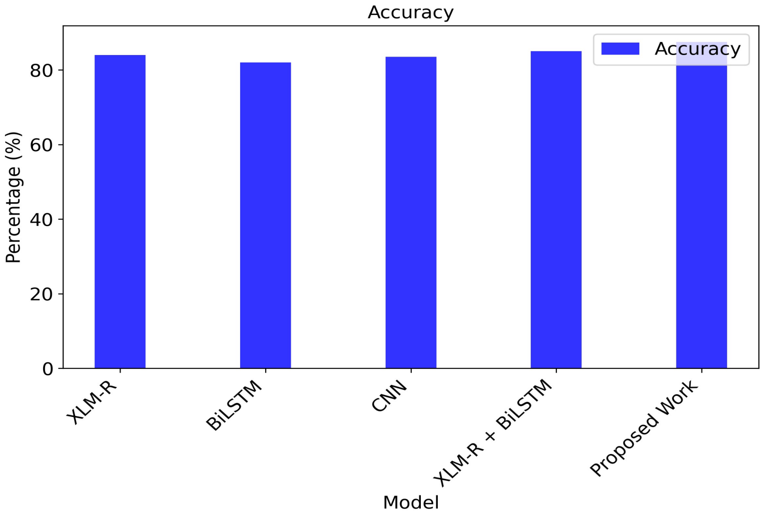

Figure 4: Comparative Analysis Of Accuracy

The AUC-ROC scores in Figure 5 here are the proposed model's highest at 91.0%. This score reflects the model's excellent ability to distinguish between the classes at various threshold settings, which indicates a highly reliable model.

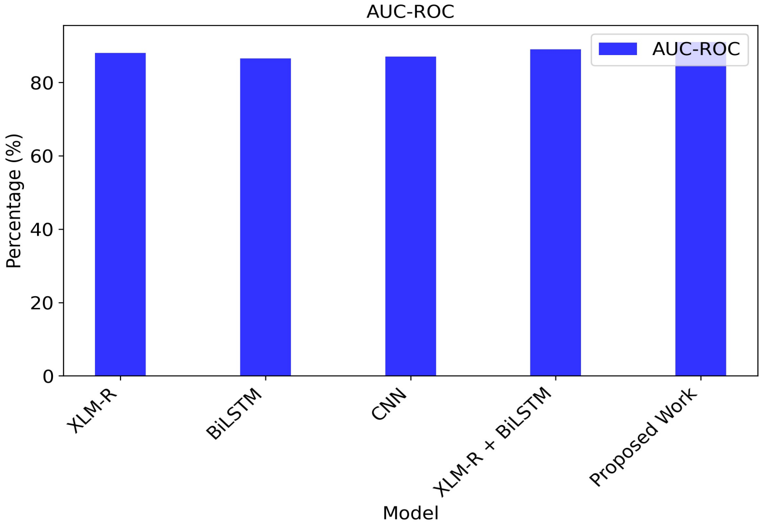

Figure 5: Comparative Analysis Of AUC-ROC

4.2 Discussion

The proposed XLM-R-BiLSTM-CNN ensemble model demonstrated substantial strengths in sentiment analysis of code-mixed texts by integrating anchor detection, robust label propagation, and a strategically layered neural architecture. The anchor detection mechanism was pivotal in utilizing unsupervised learning to identify critical data points that significantly enhance the training process. These anchors effectively captured core sentiment expressions within the dataset, providing a reliable basis for extending labels to nearby, previously unlabeled data. This method proved to be highly effective in expanding the usable training dataset without requiring extensive manual annotation, thereby enhancing the model's efficiency under semi-supervised learning conditions.

Label propagation, empowered by a novel weighting system that adjusted influence based on the proximity and density of data points around each anchor, further amplified the model’s ability to generalize from limited labeled data. This approach ensured that the propagated labels were accurate and reflective of the underlying sentiment distributions within the text, leading to improved model performance.

The ensemble architecture, combining XLM-R, BiLSTM, and CNN, leveraged the strengths of each component to address different aspects of the sentiment analysis task. XLM-R's deep contextual understanding of multiple languages provided a solid foundation for feature extraction. BiLSTM layers added the ability to capture temporal and sequential dependencies across code-switched language barriers, crucial in understanding the full context of sentiments expressed in code-mixed texts. Meanwhile, the CNN component excelled in extracting localized features and sentiment-indicative phrases often pivotal in determining the overall sentiment of short text snippets commonly found in social media.

This cohesive integration of technologies facilitated a robust analysis of complex, multilingual data, achieving superior performance compared to standard single-model approaches. The ensemble's ability to harness and synthesize diverse linguistic cues from code-mixed texts underscored its effectiveness and its potential applicability in real-world scenarios where digital communications often blend multiple languages. Thus, the proposed model represents a significant advancement in sentiment analysis, particularly in handling the nuanced challenges presented by code-mixed language datasets.

Comparison with State-of-the-Art Methods

The results of this study demonstrate that the proposed XLM-R-BiLSTM-CNN ensemble model significantly outperforms existing state-of-the-art approaches for sentiment analysis in code-mixed texts, including monolingual models and some previously proposed multilingual models. For instance, the accuracy and F1-score of the proposed model, which reached 87.5% and 86.7%, respectively, exceed the performance of Chakravarthi et al. [28], who reported F1-scores of 0.65 and 0.68 for Tamil-English and Malayalam-English sentiment analysis using traditional machine learning models like Logistic Regression and Random Forest. Furthermore, while Patwa et al. [1] achieved an F1-score of 75% using transformer-based models like BERT for code-mixed tweets, our model's superior handling of sequential dependencies and localized feature extraction allowed for improved performance.

However, some discrepancies arise when comparing our model's performance with specific domain-adaptation methods, such as Zhao et al.'s [23] work on label-efficient emotion and sentiment analysis, which reported comparable results using a transfer learning approach. Their framework demonstrated strong performance in low-resource settings, though it did not incorporate the complex ensemble architecture proposed in this paper. Additionally, while our model shows improvements in accuracy, its performance on datasets with highly imbalanced classes could still benefit from further optimization, as seen in methods like those proposed by Wang et al. (2016) in graph-based semi-supervised learning, which excel at handling such imbalance through efficient anchor selection. While the proposed model sets new performance benchmarks in handling code-mixed texts, certain limitations, particularly in extremely low-resource settings and class imbalance, could be addressed further in future research.

5. CONCLUSION

The research presented in this paper successfully demonstrates the efficacy of the proposed XLM-R-BiLSTM-CNN ensemble model for code-mixed sentiment analysis. This model leverages the combined strengths of XLM-R for robust multilingual feature extraction, BiLSTM for capturing intricate sequential dependencies, and CNN for identifying key sentiment-driven features within local text contexts. The integration of these components within an ensemble framework has shown significant improvements in accuracy, precision, recall, and F1-score over traditional single-component models. The incorporation of novel techniques such as anchor detection and dynamic label propagation has proven particularly effective. These methods not only enhance the model's ability to utilize limited labeled data more efficiently but also enable it to capitalize on the large amounts of unlabeled data typically available in real-world settings. By identifying strong sentiment indicators within the data and propagating these labels accurately across similar texts, the model addresses one of the primary challenges in semi-supervised learning: extending the reach and reliability of available annotations.

Enhancements could be explored in optimizing the computational efficiency of the anchor detection process, refining the adaptive weighting mechanism in label propagation to reduce potential propagation errors, and expanding the model's applicability to a broader range of languages and dialects. Additionally, further investigations could be made into the model's ability to adapt to new or evolving patterns of language use, which is particularly relevant in social media and other dynamically changing platforms.

REFERENCES:

[1] Patwa, Parth, Gustavo Aguilar, Sudipta Kar, Suraj Pandey, Srinivas Pykl, Björn Gambäck, Tanmoy Chakraborty, Thamar Solorio, and Amitava Das. "Semeval-2020 task 9: Overview of sentiment analysis of code-mixed tweets." *arXiv preprint arXiv:2008.04277* (2020).

[2] Baker, W., & Sangiamchit, C. (2019). Transcultural communication: language, communication and culture through English as a lingua franca in a social network community. *Language and Intercultural Communication*, *19*(6), 471–487. https://doi.org/10.1080/14708477.2019.1606230

[3] Ibrohim, Muhammad Okky, Cristina Bosco, and Valerio Basile. "Sentiment analysis for the natural environment: A systematic review." *ACM Computing Surveys* 56, no. 4 (2023): 1-37.

[4] Oueslati, Oumaima, Erik Cambria, Moez Ben HajHmida, and Habib Ounelli. "A review of sentiment analysis research in Arabic language." *Future Generation Computer Systems* 112 (2020): 408-430.

[5] Mamta, and Asif Ekbal. "Transformer based multilingual joint learning framework for code-mixed and english sentiment analysis." *Journal of Intelligent Information Systems* 62, no. 1 (2024): 231-253.

[6] Hashmi, Ehtesham, and Sule Yildirim Yayilgan. "Multi-class hate speech detection in the Norwegian language using FAST-RNN and multilingual fine-tuned transformers." *Complex & Intelligent Systems* 10, no. 3 (2024): 4535-4556.

[7] Anidjar, Or Haim, Roi Yozevitch, Nerya Bigon, Najeeb Abdalla, Benjamin Myara, and Revital Marbel. "Crossing language identification: Multilingual ASR framework based on semantic dataset creation & Wav2Vec 2.0." *Machine Learning with Applications* 13 (2023): 100489.

[8] Rawat, Anchal, Santosh Kumar, and Surender Singh Samant. "Hate speech detection in social media: Techniques, recent trends, and future challenges." *Wiley Interdisciplinary Reviews: Computational Statistics* 16, no. 2 (2024): e1648.

[9] Vegupatti, Mani, Prasanna Kumar Kumaresan, Swetha Valli, Kishore Kumar Ponnusamy, Ruba Priyadharshini, and Sajeetha Thavaresan. "Abusive Social Media Comments Detection for Tamil and Telugu." In *International Conference on Speech and Language Technologies for Low-resource Languages*, pp. 174-187. Cham: Springer Nature Switzerland, 2023.

[10] Perera A, Caldera A (2024) Sentiment Analysis of Code-Mixed Text: A Comprehensive Review. JUCS - Journal of Universal Computer Science 30(2): 242-261. https://doi.org/10.3897/jucs.98708

[11] Shetty, Poorvi. "Natural Language Processing for Tulu: Challenges, Review and Future Scope." In *International Conference on Speech and Language Technologies for Low-resource Languages*, pp. 93-109. Springer, Cham, 2024.

[12] Lu, Yingzhou, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, and Wenqi Wei. "Machine learning for synthetic data generation: a review." *arXiv preprint arXiv:2302.04062* (2023).

[13] Jabeen, Maryyam, Wasim Hassan, and Shabbir Ahmad. "Code-Mixing and Code-Switching Problems among English Language Learners: A Case Study of the University of Sahiwal." *Jahan-e-Tahqeeq* 6, no. 3 (2023): 249-258.

[14] Ghosh, Debosree. "Unleashing Customer Insights: Harnessing Machine Learning Approaches for Sentiment Analyzing and Leveraging Customer Feedback." In *Human-Centered Approaches in Industry 5.0: Human-Machine Interaction, Virtual Reality Training, and Customer Sentiment Analysis*, pp. 265-280. IGI Global, 2024.

[15] Cero, Ian, Jiebo Luo, and John Michael Falligant. "Lexicon-Based Sentiment Analysis in Behavioral Research." *Perspectives on Behavior Science* (2024): 1-28.

[16] Lee, Andrew H., Sina J. Semnani, Galo Castillo-López, Gäel de Chalendar, Monojit Choudhury, Ashna Dua, Kapil Rajesh Kavitha et al. "Benchmark Underestimates the Readiness of Multilingual Dialogue Agents." *arXiv preprint arXiv:2405.17840* (2024).

[17] Yusuf, Aliyu, Aliza Sarlan, Kamaluddeen Usman Danyaro, Abdullahi Sani BA Rahman, and Mujaheed Abdullahi. "Sentiment Analysis in Low-Resource Settings: A Comprehensive

Review of Approaches, Languages, and Data Sources." *IEEE Access* (2024).

[18] Horvat, Marko, Gordan Gledec, and Fran Leontić. "Hybrid Natural Language Processing Model for Sentiment Analysis during Natural Crisis." *Electronics* 13, no. 10 (2024): 1991.

[19] Zhao, Zishuo, Ziyang Ma, Zhenzhou Lin, Jingyou Xie, Yinghui Li, and Ying Shen. "Source-free Domain Adaptation for Aspect-based Sentiment Analysis." In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 15076-15086. 2024.

[20] Chaturvedi, Iti, Erik Cambria, Roy E. Welsch, and Francisco Herrera. "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges." *Information Fusion* 44 (2018): 65-77.

[21] Ning, Munan, Donghuan Lu, Dong Wei, Cheng Bian, Chenglang Yuan, Shuang Yu, Kai Ma, and Yefeng Zheng. "Multi-anchor active domain adaptation for semantic segmentation." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9112-9122. 2021.

[22] Wang, Meng, Weijie Fu, Shijie Hao, Dacheng Tao, and Xindong Wu. "Scalable semi-supervised learning by efficient anchor graph regularization." *IEEE Transactions on Knowledge and Data Engineering* 28, no. 7 (2016): 1864-1877.

[23] Zhao, Sicheng, Xiaopeng Hong, Jufeng Yang, Yanyan Zhao, and Guiguang Ding. "Toward Label-Efficient Emotion and Sentiment Analysis This article introduces label-efficient emotion and sentiment analysis from the computational perspective, focusing on state-of-the-art methodologies, promising applications, and potential outlooks." *Proceedings of the IEEE* (2023).

[24] Bharathi, B., and G. U. Samyuktha. "Machine Learning Based Approach for Sentiment Analysis on Multilingual Code Mixing Text." In *FIRE (Working Notes)*, pp. 1038-1043. 2021.

[25] Mandalam, Asrita Venkata, and Yashvardhan Sharma. "Sentiment analysis of Dravidian code mixed data." In *Proceedings of the First Workshop on Speech and Language Technologies for Dravidian Languages*, pp. 46-54. 2021.

[26] Chakravarthi, Bharathi Raja, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. "Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text." *Language Resources and Evaluation* 56, no. 3 (2022): 765-806.

[27] Chakravarthi, Bharathi Raja, Ruba Priyadharshini, Sajeetha Thavareesan, Dhivya Chinnappa, Durairaj Thenmozhi, Elizabeth Sherly, John P. McCrae et al. "Findings of the sentiment analysis of dravidian languages in code-mixed text." *arXiv preprint arXiv:2111.09811* (2021).

[28] Chakravarthi, Bharathi Raja, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John P. McCrae. "Corpus creation for sentiment analysis in code-mixed Tamil-English text." *arXiv preprint arXiv:2006.00206* (2020).

[29] Srinivasan, R., and C. N. Subalalitha. "Sentimental analysis from imbalanced code-mixed data using machine learning approaches." *Distributed and Parallel Databases* 41, no. 1 (2023): 37-52.

[30] M. Sangeetha, K. Nimala. Sentiment Analysis on Code-Mixed Tamil-English Corpus: A Comprehensive Study of Transformer-Based Models, 16 October 2023, PREPRINT (Version 1) available at Research Square [https://doi.org/10.21203/rs.3.rs-3418283/v1]

[31] Kumar, Lavanya Sambath, Asha Hegde, Bharathi Raja Chakravarthi, Hosahalli Shashirekha, Rajeswari Natarajan, Sajeetha Thavareesan, Ratnasingam Sakuntharaj, Thenmozhi Durairaj, Prasanna Kumar Kumaresan, and Charmathi Rajkumar. "Overview of Second Shared Task on Sentiment Analysis in Code-mixed Tamil and Tulu." In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pp. 62-70. 2024.

[32] Jahin, Md Abrar, Md Sakib Hossain Shovon, and Muhammad Firoz Mridha. "TRABSA: Interpretable Sentiment Analysis of Tweets using Attention-based BiLSTM and Twitter-RoBERTa." *arXiv preprint arXiv:2404.00297* (2024).