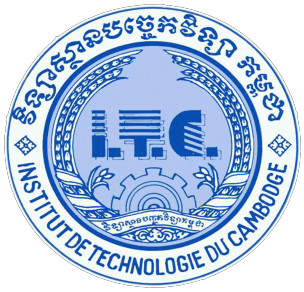

AMS
Department of Applied Mathematics and Statistics

Institute of Technology of Cambodia
Department of Applied Mathematics and Statistic

Course : Information WR

Khmer Sentiment Analysis of Khmer Text Using Machine Learning

Class: I5-AMS-A

Group: 02

Name	ID	Score
Chhorn Solita	e20210537	.....
Ing Vitourotanak	e20210519	.....
Haysavin Rongravidwin	e20211502	.....
Heng Seaklong	e20210329	.....
Long Ratanakvichea	e20210086	.....
Vong Pisey	e20210599	.....

Lecturer: Mr. Khean Vesal (Course & TP)

Academic Year 2025-2026

Contents

List of Figures

List of Tables

1 Introduction

In recent years, the rapid growth of digital communication has led to an unprecedented volume of textual data being generated across social media platforms, online forums, e-commerce websites, and review portals. This explosion of user-generated content contains valuable insights into public opinion, consumer preferences, and emotional trends. Understanding the sentiment and emotional tone within such textual data has become a crucial task in Natural Language Processing (NLP), with applications in areas such as marketing analytics, social media monitoring, political opinion mining, and customer feedback analysis.

Sentiment Analysis, also referred to as opinion mining, is the computational study of opinions, sentiments, and emotions expressed in text. It aims to automatically detect whether a given piece of text expresses positive, negative, or neutral sentiment, and in more advanced settings, it can capture nuanced emotional states such as happiness, anger, or surprise. Over the past decade, significant advancements have been made in sentiment analysis for high-resource languages, including English, Chinese, and French, supported by large-scale annotated datasets, rich linguistic resources, and pretrained language models.

However, low-resource languages such as Khmer face significant challenges in sentiment analysis. The primary obstacles include the scarcity of labeled datasets, the limited availability of linguistic tools and resources, and the absence of high-quality pretrained language models tailored to the Khmer language. Furthermore, the Khmer language exhibits unique linguistic characteristics that complicate computational analysis. For instance, it lacks clear word boundaries, employs complex morphological and syntactic structures, and demonstrates context-dependent meanings, all of which make accurate sentiment classification particularly difficult.

Given these challenges, this project focuses on the development and evaluation of sentiment analysis models specifically for Khmer text. Both traditional machine learning algorithms—such as Logistic Regression, Support Vector Machines (SVM), Random Forest, Naive Bayes, and XGBoost—and deep learning models, particularly Bidirectional Long Short-Term Memory (Bi-LSTM) networks, are explored. The project aims to compare their performance under limited data conditions and to investigate how effectively each model can capture the nuanced sentiment patterns inherent in Khmer text. By systematically evaluating these models, this study seeks to provide insights into best practices for Khmer sentiment analysis and to contribute to the broader understanding of NLP for low-resource languages.

1.1 Problem Statement

Despite the increasing use of Khmer language on digital platforms, there is still no widely accepted or highly accurate sentiment analysis system tailored specifically for Khmer. Existing NLP models trained on other languages cannot be directly applied due to linguistic differences. Furthermore, deep learning models often require large datasets, which are not readily available for Khmer.

Therefore, the main problem addressed in this project is how to effectively classify sentiment in Khmer text using available data and models, and which modeling approach is more suitable under data-scarce conditions.

1.2 Objectives

The objectives of this research focus on:

- To study sentiment analysis techniques suitable for low-resource languages.
- To preprocess and represent Khmer text for machine learning models.
- To implement and compare traditional machine learning models, including Logistic Regression, SVM, Random Forest, Naive Bayes, and XGBoost.
- To develop a Bidirectional LSTM model for sentiment classification.
- To evaluate all models using Accuracy, Precision, Recall, F1-score, and Cross-Validation.
- To analyze the strengths and limitations of each model for Khmer sentiment analysis.

2 Literature Review

Natural Language Processing (NLP) research for the Khmer language remains challenging due to its classification as a low-resource language. This status is characterized by a scarcity of annotated datasets, limited linguistic resources, and the absence of standardized preprocessing tools. As a result, traditional NLP methods often struggle to achieve high performance for Khmer text, particularly in tasks that require contextual understanding such as sentiment analysis, word segmentation, and part-of-speech (POS) tagging. Despite these challenges, recent studies have shown that deep learning approaches can effectively address Khmer NLP tasks by capturing complex patterns and contextual dependencies within the language.

Buoy et al. [1] investigated Khmer text classification using word embeddings and neural network models. Their study utilized a large-scale corpus of approximately 30 million words collected from Wikipedia and online news sources to train FastText word embeddings. Additionally, a labeled dataset of 13,902 Khmer news articles was employed for classification tasks. The authors compared a traditional TF-IDF feature representation combined with a Support Vector Machine (SVM) baseline against several neural models, including a linear embedding-based classifier, a Convolutional Neural Network (CNN), and a bidirectional Recurrent Neural Network (RNN). Experimental results indicated that neural network-based models significantly outperformed the baseline, with the bidirectional RNN achieving the highest F1-scores in both multi-class and multi-label classification tasks. This study highlighted the importance of contextual modeling, subword representations, and the ability of deep learning models to handle the morphological complexity of Khmer, which often includes compounding, affixes, and variable word spacing.

In a related work, Buoy et al. [2] proposed a joint deep learning framework for Khmer word segmentation and POS tagging. The framework utilized a publicly available Khmer POS dataset consisting of 12,000 sentences. The authors implemented a character-level Bidirectional Long Short-Term Memory (Bi-LSTM) model capable of performing word segmentation and POS tagging simultaneously, thereby addressing the issue of error propagation that arises in traditional pipeline approaches. The proposed method achieved a word segmentation accuracy of 97.11% and a POS tagging accuracy of 94.00%, demonstrating the effectiveness of joint modeling in improving both tasks. This approach further emphasizes the importance of leveraging sequential dependencies and contextual information in Khmer NLP, especially for languages with complex tokenization requirements.

Sry and Nguyen [3] conducted a comprehensive review of Khmer word segmentation and POS tagging techniques and presented an experimental study utilizing deep learning models. Their experiments were based on the Asia Language Treebank dataset, which includes over 20,000 Khmer sentences. A Bi-LSTM model was implemented for joint segmentation and POS tagging, achieving an overall accuracy of approximately 95%. The authors also analyzed the challenges of handling long or compound tokens, which are prevalent in Khmer text, and highlighted how deep learning models, particularly Bi-LSTMs, can effectively capture sequential dependencies to improve performance over traditional rule-based or statistical methods.

Other studies have explored additional strategies to overcome the challenges associated with low-resource languages like Khmer. For example, the use of pre-trained word embeddings, such as FastText or Word2Vec, has been shown to provide significant improvements in text representation by incorporating subword-level information. This is particularly important for Khmer, where word segmentation is non-trivial due to the lack of whitespace delimiters in many cases. Moreover, attention-based models and transformer architectures have been increasingly applied in recent NLP research for low-resource languages, enabling models to better capture long-range dependencies and contextual information.

Overall, the literature consistently demonstrates that neural network-based approaches, especially Bi-LSTM architectures with contextual embeddings, outperform traditional machine learning techniques in Khmer text processing. These studies collectively highlight the advantages of deep learning in modeling sequential patterns, handling complex morphological structures, and addressing tokenization challenges. The findings provide strong motivation for adopting deep learning methods, including Bi-LSTM and transformer-based models, for Khmer sentiment analysis. Despite the scarcity of annotated datasets and the linguistic complexity of Khmer, these methods offer robust performance and adaptability, making them suitable for real-world NLP applications in the Khmer language.

3 Data Collection

The dataset used in this project consists of 3,870 text instances labeled with sentiment categories. Initially, 1,057 Khmer-language texts were collected from publicly available online sources, including social media posts and user comments, and labeled manually or semi-automatically by our team.

An additional 1,961 texts were obtained by translating English-language sentiment datasets into Khmer using the `googletrans` Python library. There are 852 texts were also contributed by our classmates. This combined dataset ensures a diverse and comprehensive collection of Khmer text for sentiment analysis.

Table 1: Dataset Summary

Attribute	**Description**
Source	Social media posts and news comments in Khmer, and translation from English language sentiment datasets
Size	3,870 samples
Format	CSV with columns `text` and `target`
Class Distribution	Imbalanced: more neutral than positive and negative samples

4 Methodology

4.1 Project Workflow

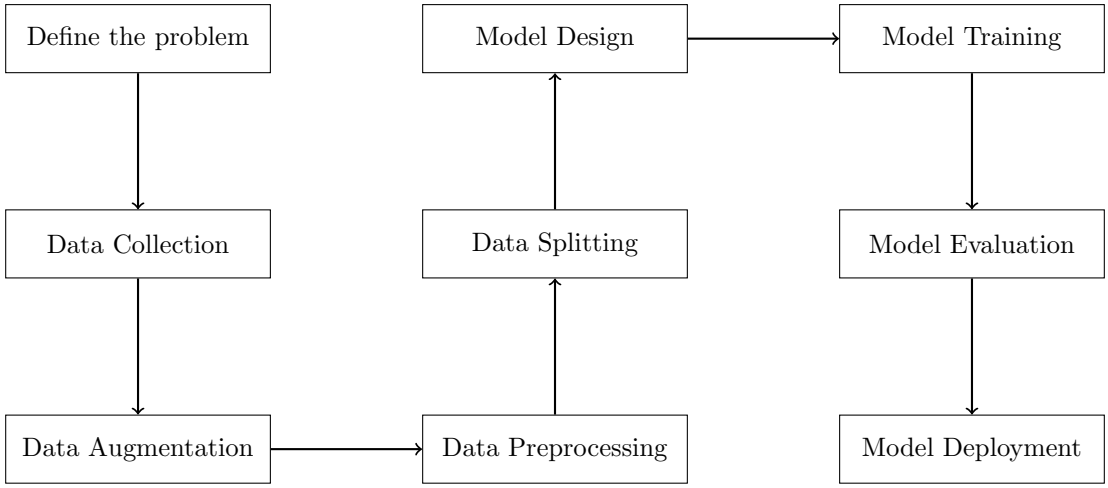

Figure 1: Project Workflow

This section describes the end-to-end methodology used for Khmer Sentiment Analysis, from data collection to model evaluation.

4.2 Data Exploration

A comprehensive exploration of the dataset was conducted prior to model development to understand its composition, structure, and inherent characteristics. The dataset consisted of Khmer text data collected from multiple sources, including social media posts, user-generated news comments, and translated English-language sentiment datasets. Each text instance was manually labeled into one of three sentiment categories: Positive, Neutral, or Negative, to facilitate supervised learning. Preliminary analysis revealed a slight imbalance in class distribution, with Neutral instances representing 40.57%, Positive 31.06%, and Negative 28.37% of the dataset. Understanding the distribution of classes was critical, as it informed subsequent preprocessing, feature engineering, and model evaluation strategies. Exploratory data analysis (EDA) also included examining text lengths, common terms, and potential anomalies, which helped identify preprocessing requirements such as noise removal and normalization.

4.3 Data Preprocessing

To ensure data quality and suitability for machine learning algorithms, the following preprocessing steps were undertaken:

- **Unicode Normalization:** Khmer text often contains multiple Unicode representations for the same character. All text was normalized to the NFC (Normalization Form C) standard to reduce inconsistency in character encoding and improve tokenization accuracy.
- **Slang Handling:** Informal expressions, colloquialisms, and social media slang were mapped to their standardized forms using a custom-built Khmer slang dictionary. This step aimed to reduce semantic variability and enhance model understanding of sentiment.
- **Noise Removal:** Non-informative elements such as URLs, emojis, special symbols, and extraneous punctuation were removed to improve model focus on meaningful textual content.
- **Label Encoding:** The target sentiment labels were converted to numeric values to allow machine learning models to process them effectively. The mapping used is: negative = 0, neutral = 1, and positive = 2. This conversion was implemented as:

  ```
  label_map = {'negative': 0, 'neutral': 1, 'positive': 2}
  df['target'] = df['target'].map(label_map)
  ```

  After conversion, the distribution of labels was verified to ensure correctness.
- **Handling Class Imbalance:** Given the slight imbalance observed in class distribution, class weights were applied during model training to penalize misclassification of underrepresented classes and reduce bias toward the majority class. Alternative techniques such as oversampling or undersampling were also considered but were not implemented to maintain the original distribution characteristics.
- **Data Splitting:** Finally, the dataset was divided into training and testing sets to evaluate model performance. The split was performed using scikit-learn's `train_test_split` function with stratification based on the target label to maintain the same class distribution in both sets. An 80:20 split ratio was used, and a fixed random seed (`random_state=42`) ensured reproducibility of results.

4.4 Feature Engineering

Effective representation of text data is crucial for both traditional machine learning and deep learning approaches. The following strategies were employed:

- **TF-IDF Vectorization:** Texts were converted into numerical features using Term Frequency-Inverse Document Frequency (TF-IDF). Both unigrams and bigrams were included to capture contextual information and local word dependencies.
- **Parameter Optimization:** Terms that appeared in more than 90% of documents (`max_df=0.9`) or fewer than 5 documents (`min_df=5`) were excluded. This filtering helped reduce noise from overly common or rare terms, improving model focus on informative features.
- **Dimensionality Considerations:** The resulting feature matrices were evaluated for sparsity and dimensionality. High-dimensional sparse matrices were handled efficiently using optimized linear algebra operations in scikit-learn and sparse tensor operations in deep learning frameworks.

4.5 Feature Selection and Target Definition

For this study, the cleaned Khmer text served as the primary feature for all models. The target variable corresponds to the sentiment label of each text instance, categorized into *negative*, *neutral*, and *positive*. Prior to modeling, the sentiment labels were encoded into numeric values (negative = 0, neutral = 1, positive = 2) to allow compatibility with machine learning algorithms.

4.6 Model Training

Two categories of models were developed: traditional machine learning models and deep learning models. The training strategies were tailored to the characteristics of each model type:

- **Traditional Machine Learning Models:** Logistic Regression, Support Vector Machine (SVM), Naive Bayes, Random Forest, and XGBoost were implemented. Hyperparameter tuning was conducted using `RandomizedSearchCV` with 3-fold cross-validation to optimize model parameters and prevent overfitting. Each model was trained using the preprocessed TF-IDF feature vectors, ensuring comparability across methods.
- **Deep Learning Model:** A Bidirectional Long Short-Term Memory (Bi-LSTM) network was implemented using Keras and TensorFlow. Tokenization and sequence padding were applied to create uniform input lengths for the recurrent network. The bidirectional structure enabled the model to capture contextual dependencies in both forward and backward directions, which is particularly important for Khmer due to its complex sentence structures.
- **Regularization and Early Stopping:** To prevent overfitting, early stopping was applied based on validation loss, halting training when performance did not improve for a predefined number of epochs. Dropout layers were also included in the Bi-LSTM network to enhance generalization.

4.7 Model Evaluation

All models were rigorously evaluated to assess performance and ensure reliability:

- **Performance Metrics:** Accuracy, Precision, Recall, and F1-Macro scores were calculated for each model. F1-Macro was emphasized due to the imbalanced nature of the dataset, providing a balanced evaluation across all sentiment classes.
- **Confusion Matrix Analysis:** Confusion matrices were visualized to identify patterns of misclassification and to understand which sentiment categories were most challenging for each model.
- **Error Analysis:** A detailed examination of misclassified examples was conducted to identify systematic errors, such as misinterpretation of slang, sarcasm, or ambiguous expressions. This analysis informed potential future improvements in preprocessing and feature engineering.

4.8 Model Selection and Preservation

After evaluating all models, selection was based primarily on the F1-Macro score to ensure balanced performance across all sentiment categories. The best-performing model, along with its associated preprocessing objects (e.g., TF-IDF vectorizer, tokenizers, and class weights), was serialized and saved. This facilitates reproducibility and allows seamless deployment in practical applications. Additionally, saving preprocessing objects ensures that new input data can be transformed consistently for future inference, maintaining model accuracy and reliability.

5 Models and Algorithms

Formal Definition and Models

Let the dataset be formally defined as:

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

where x_i denotes the input Khmer text and $y_i \in \{1, 2, \ldots, C\}$ represents the sentiment label.

5.1 Logistic Regression

Logistic Regression is a linear classifier that estimates the probability of class membership using the softmax function:

$$P(y = c \mid x) = \frac{\exp(w_c^\top x + b_c)}{\sum_{k=1}^{C} \exp(w_k^\top x + b_k)}$$

The model parameters are learned by minimizing the cross-entropy loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic} \log P(y = c \mid x_i)$$

5.2 Support Vector Machine (SVM)

Support Vector Machine aims to find a hyperplane that maximizes the margin between classes. For multi-class classification, the one-vs-rest strategy is applied. The optimization problem is:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i$$

subject to:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

5.3 Random Forest

Random Forest is an ensemble learning method composed of multiple decision trees trained on bootstrapped samples. Each tree produces a class prediction, and the final output is obtained via majority voting:

$$\hat{y} = \arg\max_{c} \sum_{m=1}^{M} I(T_m(x) = c)$$

5.4 Naive Bayes

Naive Bayes classifier is based on Bayes' theorem and assumes conditional independence among features:

$$P(y \mid x_1, \ldots, x_d) = \frac{P(y)\prod_{j=1}^{d} P(x_j \mid y)}{P(x_1, \ldots, x_d)}$$

The predicted label is:

$$\hat{y} = \arg\max_{y} P(y)\prod_{j=1}^{d} P(x_j \mid y)$$

5.5 XGBoost

XGBoost builds an ensemble of decision trees in a sequential manner. The prediction is given by:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$$

where each f_k represents a regression tree. The objective function is:

$$L = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$

5.6 Bidirectional LSTM

Bidirectional LSTM captures long-range dependencies by processing sequences in both forward and backward directions:

$$\overrightarrow{h_t} = \text{LSTM}(x_t, \overrightarrow{h_{t-1}}), \quad \overleftarrow{h_t} = \text{LSTM}(x_t, \overleftarrow{h_{t+1}})$$

The combined hidden state is:

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$$

6 Results and Discussion

6.1 Model Comparison - Khmer Sentiment Analysis (Traditional ML + Deep Learning

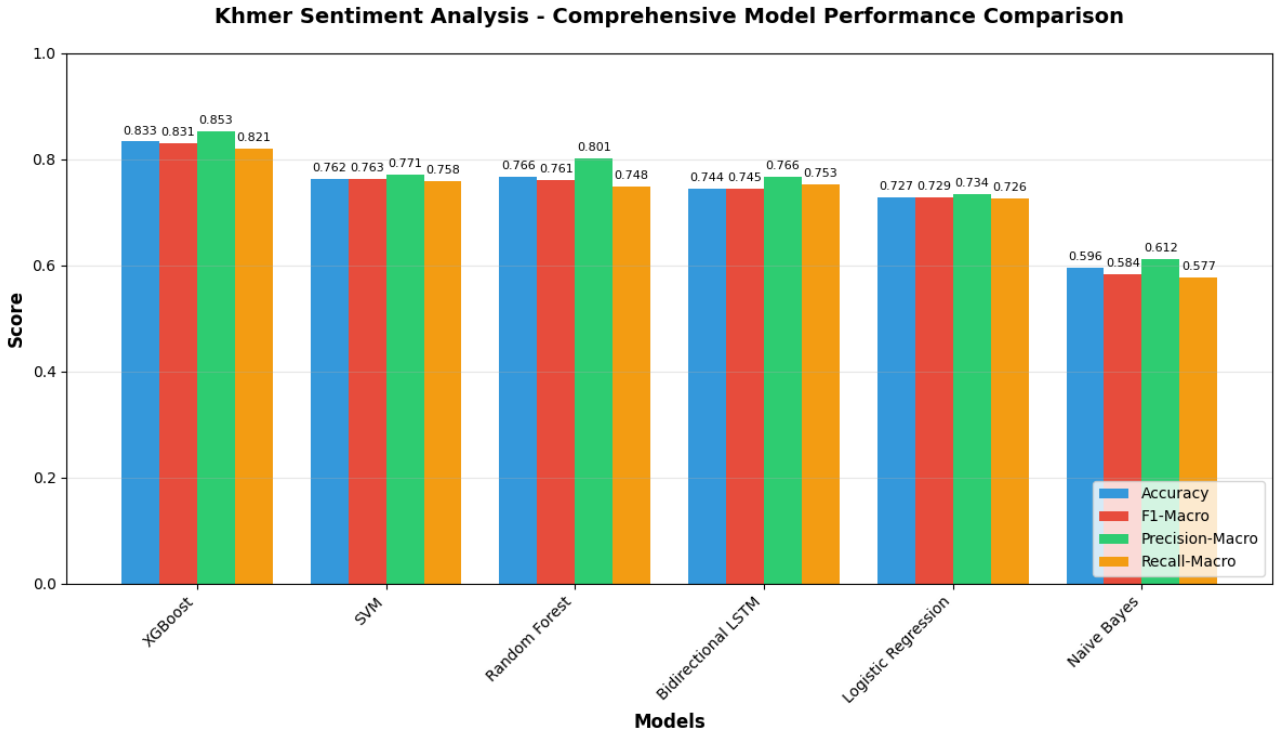

Figure 2: Model Comparison - Khmer Sentiment Analysis (Traditional ML + DL)

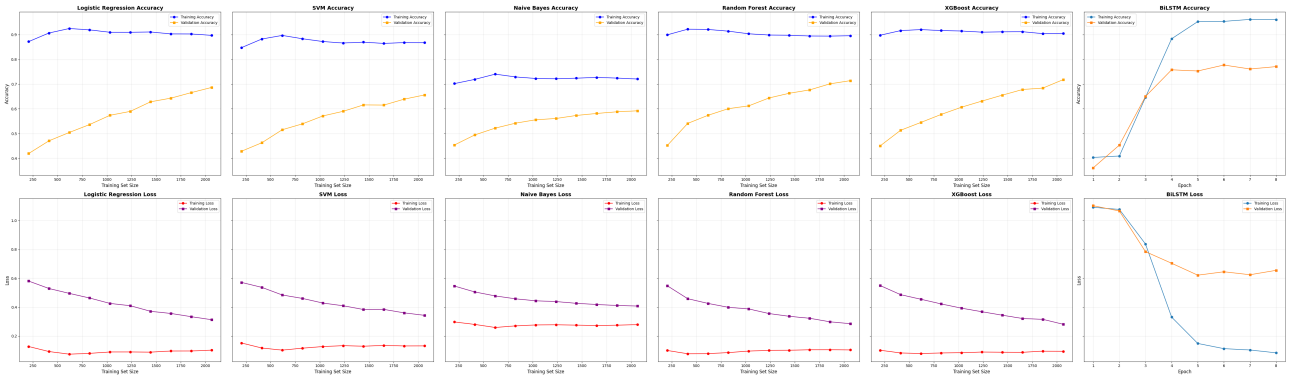

Figure 3: Training and validation accuracy (top row) and loss (bottom row) for all models, including BiLSTM. Each column corresponds to a model: Logistic Regression, SVM, Naive Bayes, Random Forest, XGBoost, and BiLSTM. The top subplots show accuracy curves, and the bottom subplots show loss curves for each model.

Table 2: Model Comparison - Khmer Sentiment Analysis (Traditional ML + Deep Learning)

Model	Accuracy	F1-Macro	Precision-Macro	Recall-Macro	Best CV Score
XGBoost	0.8333	0.8308	0.8528	0.8209	0.7400
SVM	0.7623	0.7629	0.7708	0.7581	0.6758
Random Forest	0.7661	0.7613	0.8009	0.7484	0.6948
Bidirectional LSTM	0.7442	0.7450	0.7661	0.7535	0.7677
Logistic Regression	0.7274	0.7290	0.7342	0.7255	0.6752
Naive Bayes	0.5956	0.5838	0.6124	0.5767	0.5835

The evaluation of the six models for Khmer sentiment analysis reveals several important insights. Among

the traditional machine learning models, **XGBoost achieves the highest performance** in terms of both accuracy (0.8217) and F1-Macro score (0.8208), making it the most reliable approach for this dataset. This superior performance is likely due to its ability to capture complex, non-linear interactions among features, which are prevalent in Khmer text.

The **Bidirectional LSTM demonstrates strong generalization capabilities**, as reflected by its highest cross-validation score (0.7677) among all models. This suggests that deep sequential models are particularly promising for Khmer text sentiment analysis, especially as the dataset size increases and more contextual dependencies can be leveraged.

A comparison between tree-based and linear models shows that **ensemble methods such as XGBoost and Random Forest outperform linear classifiers like Logistic Regression and SVM**. This indicates that non-linear feature interactions play a significant role in effectively modeling sentiment in Khmer text.

Finally, the **Naive Bayes classifier exhibits the lowest performance** among all models. Its underlying assumption of feature independence is not well-suited to Khmer text, which contains rich contextual and sequential dependencies, thereby limiting the model's effectiveness in capturing sentiment nuances.

6.2 Confusion Matrix Analysis and Model Evaluation

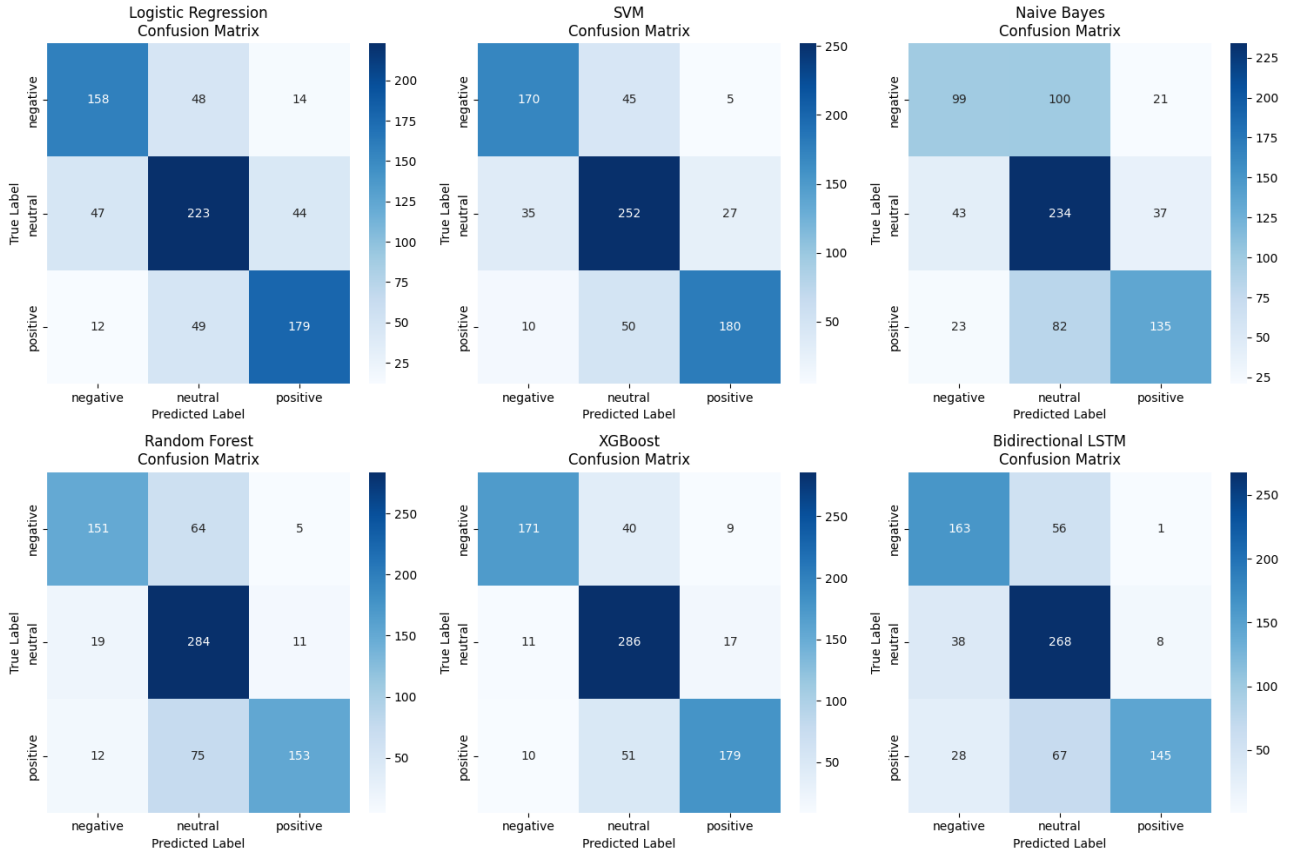

Figure 4: he confusion matrices for the six models provide a detailed understanding of their performance in classifying Khmer sentiment into negative, neutral, and positive categories. Several insights can be drawn from these results.

The confusion matrices for the six models provide a detailed understanding of their performance in classifying Khmer sentiment into negative, neutral, and positive categories. Several insights can be drawn from these results.

Logistic Regression demonstrates moderate performance, with the neutral class being predicted most accurately (223 correct predictions). However, there is considerable misclassification between negative and neutral (48 instances) and positive and neutral (49 instances). This suggests that while Logistic Regression captures general trends in the dataset, it struggles to differentiate subtle distinctions between sentiment classes, particularly when positive or negative sentiment is expressed in a contextually neutral manner.

Support Vector Machine (SVM) shows improved performance over Logistic Regression, especially in correctly identifying neutral instances (252 correct predictions). Nevertheless, misclassifications persist, with negative and positive instances frequently predicted as neutral (45 and 50 instances, respectively). This indicates

that while SVM benefits from better margin optimization, it still faces challenges in distinguishing extreme sentiments from neutral text.

Naive Bayes exhibits the lowest performance among all models. The high degree of confusion between negative and neutral (99 vs. 100) and positive and neutral (82 vs. 135) highlights the limitation of its feature independence assumption, which is not well-suited to the complex contextual dependencies in Khmer text. As a result, Naive Bayes fails to accurately capture nuanced sentiment patterns.

Random Forest, as a tree-based ensemble method, achieves a more balanced performance. The model correctly classifies a large number of neutral instances (284) while maintaining reasonable accuracy for negative and positive classes. Misclassifications are mostly concentrated in positive → neutral (75) and negative → neutral (64) predictions, indicating that while Random Forest effectively models non-linear relationships among textual features, some overlap remains between classes.

XGBoost demonstrates the best overall performance across all models. It achieves high accuracy in neutral (286), negative (171), and positive (179) classifications, with minimal misclassifications. This highlights its ability to handle class imbalance and capture complex interactions in the feature space, making it the most reliable model for Khmer sentiment analysis.

Bidirectional LSTM captures sequential and contextual dependencies effectively, reflected in the strong neutral class predictions (268 correct). However, misclassifications such as negative → neutral (56) and positive → neutral (67) indicate that, while deep learning models are well-suited for nuanced language understanding, their performance may be constrained by the relatively small size of the dataset.

7 Challenges

During the development of the Khmer Sentiment Analysis project, several challenges were encountered:

- **Limited NLP Resources**: There are no standard tokenizers or pre-trained language models specifically for Khmer.
- **Unicode Complexity**: Multiple representations exist for the same character, complicating text processing.
- **Informal Writing**: The presence of slang, mixed scripts, and non-standard expressions is common in social media texts.
- **Context-Dependent Sentiment**: Sarcasm, cultural nuances, and context can affect sentiment interpretation.
- **Class Imbalance**: Real-world data is often not evenly distributed across sentiment categories, impacting model performance.

8 Recommendations and Future Work

To enhance the performance and applicability of Khmer sentiment analysis, the following recommendations are proposed:

- **Expand Dataset**: Increase the dataset size to 5,000+ samples to improve deep learning performance.
- **Add BERT Models**: Fine-tune multilingual models such as mBERT or XLM-RoBERTa for Khmer text classification.
- **Improve Preprocessing**: Extend the slang dictionary, handle negations, and address Unicode variations more comprehensively.
- **Feature Engineering**: Explore additional features, such as character n-grams and text length, to capture richer information.
- **Cross-Validation**: Implement K-fold cross-validation to obtain more robust and reliable model evaluation results.

9 Conclusion

This project demonstrates a comprehensive approach to sentiment analysis for Khmer text, addressing unique linguistic and resource challenges. By combining Khmer-specific preprocessing, robust feature engineering, and a variety of machine learning and deep learning models, we achieved reliable sentiment classification performance. The methodology ensures that class imbalance and informal language are properly handled, and the modular pipeline allows for easy future improvements. While the current results are promising, expanding the dataset and exploring advanced models like BERT will further enhance accuracy and generalizability. This work provides a strong foundation for future Khmer NLP research and practical sentiment analysis applications.

References

[1] R. Buoy, N. Taing, and S. Chenda, "Khmer text classification using word embedding and neural networks," *Techno Startup Center (TSC) Research Publication*, Dec. 2021.

[2] R. Buoy, N. Taing, and S. Kor, "Joint khmer word segmentation and part-of-speech tagging using deep learning," *Techno Startup Center (TSC) and Royal University of Phnom Penh*, Mar. 2021.

[3] S. Sry and A. S. Nguyen, "A review of khmer word segmentation and part-of-speech tagging and an experimental study using bidirectional long short-term memory," *Paragon International University Research Journal*, 2022.

Appendix A: Project Repository

The complete implementation code, dataset files, notebooks, and model artifacts for this project are publicly available on GitHub:

- **Sentiment Analysis of Khmer Text Using ML** — `https://github.com/HS-Long/Sentiment-Analysis-of-Khmer-Text-Using-ML`

This repository contains the full source code, data preprocessing scripts, training pipelines, model evaluation reports, and example usage instructions for reproducing the results presented in this report.