Unravelling Emotional Tones: A Hybrid Optimized Model for Sentiment Analysis in Tamil Regional Languages

[1]Sangeetha M[1] and [2]Nimala K
[1]Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, TamilNadu, India.
[2]Networking and Communication, SRM Institute of Science and Technology, Kattankulathur, TamilNadu, India.
[1]sm6281@srmist.edu.in, [2]nimalak@srmist.edu.in

Correspondence should be addressed to Nimala K : nimalak@srmist.edu.in.

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404012
Received 02 June 2023; Revised from 28 August 2023; Accepted 26 October 2023.
Available online 05 January 2024.

Abstract – Review comments from digital platform such as Facebook, Twitter and YouTube used for identification of emotional tones from text. Nowadays, reviews are posted in different languages such as English, French, Chinese, and Indian regional languages such as Tamil, Telegu, and Hindi. Identification of emotional tones from text written in Indian regional language is challenging. During the translation of the regional language to the English language for sentiment analysis, lexical and pragmatic ambiguity are the major problem. The above problem arises due to dialects in language such as regional, standard, and social dialects. In this paper, dialect-based ambiguity problems solve through proposed Hybrid optimized deep learning transformer Models like M-BERT, M-Roberta, and M-XLM-Roberta for Tamil language dialects recognise and classified. The proposed algorithms provide better sentimental analysis after Hybrid optimization due to adaptation mechanisms, dynamic changes in the parameters and strategies in fine-tuning the search. The proposed Hybrid optimized algorithms perform better than existing algorithms such as SVM, Naïve Bayes, and LSTM with an accuracy of 95%.

Keywords-NLP, LSTM, Dialect, Lexical Ambiguity, Hyperparameter, Fine Tuning.

I. INTRODUCTION

Natural Language processing in Indian regional languages such as Tamil, Telegu, Malayalam relies on linguistic resources such as corpus dictionaries, grammar, part-of-speech taggers, and morphological analysers. For Tamil languages, NLP systems access the vital information about Tamil's vocabulary, grammar rules, and linguistic characteristics. To build a comprehensive corpus for a regional language, such as Tamil, an extensive range of text sources including books, articles, websites and social media posts are used. Corpus reflects the diverse vocabulary, grammar, and usage patterns of the language [1]. In NLP and regional languages cleaned and pre-processed. During this stage, unnecessary characters, punctuation, and numbers are eliminated, and specific challenges such as spelling variations and abbreviations unique to the regional language are addressed. Additionally, the text is divided into individual words and sub-words for further analysis and processing [2]. In certain regional languages, words are not always separated by spaces or punctuation marks. In such cases, word segmentation is used for to identify the boundaries between words, which determines one word ends and the next one begins. Word segmentation is removed with different techniques, such as statistical models, rule-based approaches, or machine learning algorithms, accurately establish word boundaries [3]. In regional languages, the morphology of words tends to be complex, with affixes, infixes, and other morphological operations.

Morphological analysis is the process of dissecting words into constituent morphemes and determines grammatical properties. This involves techniques such as stemming, lemmatization, and part-of-speech tagging to accurately analyze and categorize the words based on their morphological characteristics. To perform NER accurately, dedicated models are trained specifically for the regional language [4][5]. Language modelling is a training model based on the grammar, context, and semantics of a particular language. In the case of regional languages, language models are trained on specific datasets to capture the shades and unique characteristics of the language [6].

Dialects are the major problem in natural language processing (NLP) due to their influence on language variation and understanding. NLP enables computers to process and understand human language.

Dialects refer to regional or social language variations characterized by differences in pronunciation, vocabulary, grammar, and other linguistic aspects. These variations emerge organically within distinct geographic or social communities. NLP encounters challenges when dealing with dialectal variations in text which lead to accuracy in NLP models such as speech recognition, language understanding, machine translation, and sentiment analysis.

Models trained on one dialect may struggle to perform effectively on data from another dialect due to variations in pronunciation, grammar, and vocabulary. The process of dialectal data plays a vital role in developing NLP models for effective handling a dialectal variation. Large and diverse datasets that encompass different dialects is crucial to enhance the performance of NLP systems across various regions and user groups. Linguists and researchers collaborate to obtain resources such as dialectal corpora, lexicons, and pronunciation dictionaries, which facilitate NLP research on dialects. Resources play a pivotal role in the development of dialect-aware models and systems for effective analysis and process of dialectal variations in NLP applications.

Accents and pronunciation reflect the variations in dialects. To ensure accurate recognition and transcription of spoken language, when dealing with diverse dialects or non-standard speech, NLP systems require training on a wide range of accent data. This helps the systems to adapt and handle the accent and pronunciation variations that occur across different dialects.

Code-switching is a practice among dialect speakers, speakers alternate between different languages or dialects during a conversation. NLP systems face the challenge of handling code-switching must seamlessly recognize and comprehend multiple dialects or languages within the same context. Addressing this challenge requires the development of robust models and techniques that can accommodate and interpret code-switching instances accurately. To enhance the performance of NLP models on specific dialects fine-tuned using dialectal data.

This process involves utilizing the data from a particular dialect and improves the model understanding and accuracy for specific dialect. Additionally, transfer learning techniques enable models trained on one dialect or language to be applied to another dialect or language with less training data, leveraging the knowledge gained from the source dialect and the enhance performance in the target dialect. NLP researchers consider sociolinguistic factors related to dialects, which include the social class, age, education level, and gender. The above factors have an impact on the variations in dialects and influence the design and effectiveness of NLP systems.

Tamil is one of the major Dravidian languages spoken in South India and Sri Lanka, with various dialects due to long history and regional variations. The dialects are broadly classified into regional dialects, standard dialects, and social dialects. Tamil regional dialects are due to variances in geography and culture among different regions.

Several notable regional dialects include :(i) Madras Tamil (Chennai Tamil): This dialect is associated with the city of Chennai and surrounding areas. It has a significant influence on modern spoken and colloquial Tamil due to Chennai's prominence in the film and media industry. (ii) Kongu Tamil: Spoken in the western region of Tamil Nadu, particularly in the districts of Coimbatore, Erode, and Tirupur. Kongu Tamil has distinct vocabulary, phonetic variations, and grammatical differences compared to other dialects.(iii) Madurai Tamil: Associated with the city of Madurai and surrounding regions, this dialect has its own unique flavour in terms of pronunciation, vocabulary, and idiomatic expressions.(iv) Jaffna Tamil: Spoken predominantly in the northern part of Sri Lanka, Jaffna Tamil has own set of vocabulary, pronunciation patterns, and grammatical distinctions compared to Tamil spoken in other regions.

Standard Dialect: The standard dialect of Tamil, known as "Centamil" or "Sankethi," is based on the dialect spoken in the city of Chennai. It serves as the formal variety used in text, education, literature, media, and official communication. The standard dialect is understood and spoken by educated Tamil speakers across different regions and provides a common linguistic base for the Tamil-speaking population. Social Dialects: Social dialects in Tamil emerge based on factors such as social class, education level, and occupation. These variations are never necessarily tied to a specific region, which are influenced by social factors. Some examples of social dialects include:

(i) Brahmin Tamil: Spoken by the Brahmin community in Tamil Nadu, this dialect incorporates distinctive vocabulary, grammar, and pronunciation patterns influenced by Brahmin culture and traditions.

(ii) Colloquial Tamil: This informal variant is commonly spoken in everyday conversations, especially among the younger generation. It often incorporates loanwords from English and other languages, along with relaxed grammar and pronunciation.

(iii) Tamil Vernaculars: Tamil vernaculars refer to the localized dialects spoken in specific occupational or community contexts. For example, the fishing community in coastal areas has a unique dialect influenced by their maritime lifestyle.

Sentiment analysis is known as opinion mining, the process of determining the sentiment or emotional tone behind a piece of text, such as a sentence, paragraph, or document. It analyses the subjective content and classify as positive, negative, or neutral. Sentiment analysis is applied to various languages, including regional languages, to gain insights into people's opinions, attitudes, and emotions. Language resources play a vital role in the development of sentiment analysis models. Resources such as labelled datasets, lexicons, and pre-trained models, are essential for accurate sentiment analysis. However, regional languages Tamil, the availability of such resources may be relatively limited compared to major languages like English. Consequently, constructing robust sentiment analysis models for regional languages like Tamil may necessitate dedicated efforts to gather and annotate large-scale datasets specifically tailored for sentiment analysis.

Lexical challenges are prevalent in sentiment analysis models; they depend on lexical resources like sentiment lexicons. These lexicons consist of words and their corresponding sentiment scores. However, developing comprehensive sentiment lexicons for regional languages is intricate due to the diverse range of dialects, slang, and colloquial expressions found within the language. The creation and refinement of lexicons for regional languages often necessitate domain-specific knowledge and linguistic expertise captures the nuanced sentiment variations accurately. Regional languages exhibit a significant level of morphological complexity characterized by intricate inflections and word formations. This complexity is a challenge in sentiment analysis models, particularly if they are not specifically designed to handle morphologically rich languages. Effectively managing morphological variations becomes imperative to achieve accurate sentiment classification in regional languages. For sentiment analysis models needs contextual understanding and sensitivity to the nuances of language. Regional languages often incorporate cultural references, idiomatic expressions, and local sentiments, never adequately captured by standard sentiment analysis resources. Adapting sentiment analysis models to regional languages necessitates a deep comprehension of the cultural and linguistic context unique to that particular language.

II. LITERATURE SURVEY

There has been a big impact on NLP from large-scale pre-trained language models like GPT-3 and T5. The models have achieved state-of-the-art performance on various tasks by learning representations from massive amounts of unlabelled text. In Transformer-Based Architecture, Transformers is introduced by the "Attention Is All You Need" and becomes the central architecture in NLP. Transformers are effective in tasks like machine translation, text generation, question answering, and more. In Multilingual NLP, develop models and techniques that can understand and generate text in multiple languages. Research in this area includes cross-lingual transfer learning, multilingual embedding, and zero-shot learning. In Few-Shot and Zero-Shot Learning, Researchers explore techniques and new tasks[7][8] with limited or no labelled data. Few-shot learning adapts models quickly with few examples, while zero-shot learning enables models to perform unseen tasks by leveraging transfer learning. In Ethical and Bias Considerations, the NLP community has become more attentive to ethical concerns and biases present in language models and datasets [9].

Research is focused on addressing bias in NLP systems, developing fairness metrics, and promoting responsible practices. In NLP for Low-Resource Languages, NLP techniques apply in languages with limited resources and linguistic diversity for developing techniques for low-resource machine translation, sentiment analysis, and named entity recognition. In Neural Architecture Search, Researchers automate the design of neural network architectures for NLP tasks [10]. Neural architecture search (NAS) techniques discover optimal architectures for specific tasks and lead to efficient and effective models. In Interpretability and explain ability, Researchers have investigated techniques to make NLP models more transparent and interpretable. This includes attention visualization, saliency analysis, and understanding model decisions and biases. In NLP in Specific Domains, studies have focused on NLP applications in specific domains, such as healthcare, finance, legal, and customer support. These domain-specific NLP models aim to understand and process text within specialized contexts.

Continual Learning and Lifelong NLP, develops models that learn from new data over time, retaining knowledge from previous tasks while adapting to new ones. Lifelong NLP research aims to build models which learn incrementally and perform well on range of tasks without catastrophic forgetting. In Dialect Identification and Classification, Researchers explore techniques too automatically and identify classification of different dialects of Tamil, develop models to differentiate regional variations, and understand the specific linguistic features associated with each dialect. In Dialectal Variation in Sentiment Analysis and Opinion Mining, dialectal variations in sentiment and opinion analysis are performed in relevant areas of research. This involves understanding how sentiment expressions and opinions differ across Tamil dialects and developing methods to account for dialectal variations in sentiment analysis tasks. In Machine Translation and Tamil Dialects, Research is focused on machine translation systems specifically tailored to handle dialectal variations in Tamil. This includes exploring approaches to capture and preserve dialect-specific nuances during translation between Tamil dialects or between Tamil and other languages. In Named Entity Recognition (NER) in Tamil Dialects NLP, develop NER systems, which handle the variation of named entities across different Tamil dialects. This involves building resources, annotated datasets, and models specific to dialectal variations in Tamil.

In Code-Switching and Tamil Dialects, Code-switching is a linguistic phenomenon and the study of the code-switching patterns in Tamil dialects is of interest. Researchers develop models and techniques to understand, analyze, and process code-switched text involving Tamil dialects. In Dialectal Variations in Tamil Language Generation, Researchers explore dialect-aware language generation models for Tamil [11] characteristics, such as grammar, vocabulary, and cultural references]. This involves developing techniques and generating text that captures dialect-specific Dialect Transfer Learning, leverages knowledge from one dialect and improves performance in another dialect [12]. Researchers apply transfer learning techniques that effectively utilize data and resources from one Tamil dialect and enhance NLP models for other Tamil dialects. In Dialectal Resources and Corpus Development, the creation of dialect-specific resources and annotated corpora plays a crucial role in NLP research [13]. Comprehensive linguistic resources, such as lexicons, annotated datasets, and dialect-specific language models, are used in NLP-based Tamil dialects.

Statistical Inference from Literature Survey

The performance of the Machine learning and Deep learning algorithms in Tamil text classification tasks is as in **Table 1.**

Table 1. Performance Of ML And DL In Tamil Dialects

Classifier	Accuracy (%)	Error in translation from Tamil to English	Inference
SVM [7]	78.7	English: "I'm going to the market." Chennai Tamil: "Market kupoittuvaren."	Chennai Tamil-based reviews and translation in accuracy occurs
RNN with LSTM [10]	79.1		
M-BERT	82.2		
GBA(XGBoost, LightGBM) [8]	79.4	English: “What are you doing?” Standard Tamil: "Nee ennaseirai?" Kongu Tamil: "Neengaennapanreenga?"	Kongu Tamil-based reviews and translation in accuracy occurs
Random Forest [11]	79.4		
M-BERT	84.2		
CNN [9]	81.3	Standard Tamil: "Innorukudumbakootamvandhuruchu." Madurai Tamil: "Orukudumbakoothamvanthuruken."	Madurai Tamil-based reviews and translation in accuracy occurs
M-BERT	92.85		

The Analysis of Various Classifiers algorithms on the basis of Accuracy for classification of Tamil text as in **Fig 1.**

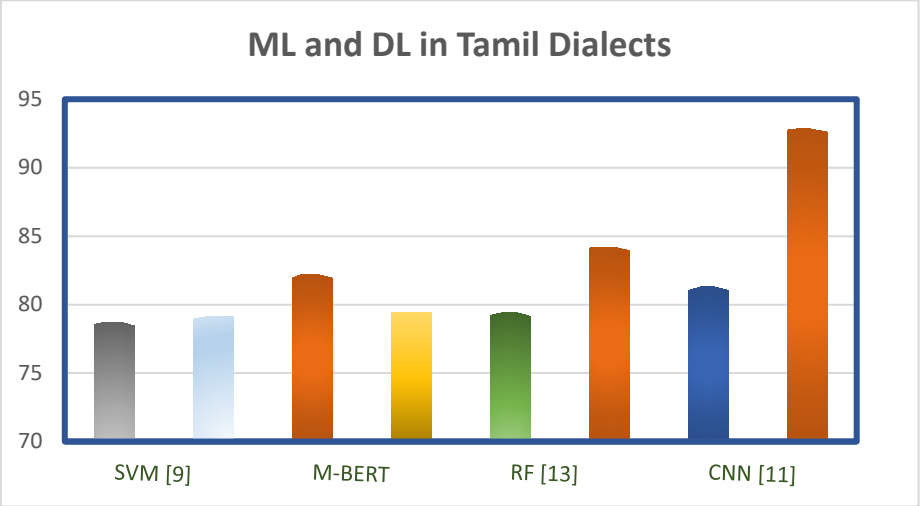

Fig 1. Analysis of Various Classifiers algorithms on the basis of Accuracy for classification of Tamil text

From YouTube reviews, feedback from viewers plays a crucial role in shaping the success of products and services. Many platforms utilize sentiment analysis techniques to gauge the sentiment expressed from reviews and relyon analysis for decision-making, such as awarding prizes and allocating resources [14]. When dealing with multilingual YouTube reviews, the challenge of lexical ambiguity becomes increasingly complex, compounded by the presence of pragmatic ambiguity. This article explores the problem of effectively analysing sentiment in multilingual YouTube reviews and considers lexical and pragmatic ambiguity during classification and sentiment analysis.

III. METHODOLOGY

In experimentation and evaluation, proposed model demonstrates significant improvements in accurately identifying sentiment in dialect YouTube reviews. By incorporating techniques to handle lexical ambiguity and leveraging contextual information to tackle pragmatic ambiguity, model performance better compared to existing approaches. The results indicate the effectiveness of our approach in capturing the intended sentiment expressed by users across different languages, thereby enables more reliable decision-making based on sentiment analysis of multilingual YouTube reviews. The overall methodologies for dialect-based classification of text as shown in **Fig 2**.

YouTube Reviews in the Tamil Language

Tamil comments on YouTube refer to comments left by users in the Tamil language on YouTube videos. The Tamil language is a widely spoken language in the southern Indian state of Tamil Nadu and the Tamil language spread around the world. YouTube allows users to leave comments on videos, express their thoughts, opinions, and feedback, or engage

in discussions related to the video content. Tamil-speaking users leverage this feature to interact with the YouTube community Tamil language. The comments are a wide range of topics such as entertainment, education, news, music, cooking, technology, and more. Tamil YouTube comments reflect the diversity of the Tamil-speaking community, including their cultural, social, and linguistic nuances. Users' express appreciation for the video, share personal experiences, ask questions, offer suggestions, or engage in conversations with other users. Tamil YouTube comments play a significant role in fostering community engagement and, allow viewers to connect with content creators and other users, who share similar interests.

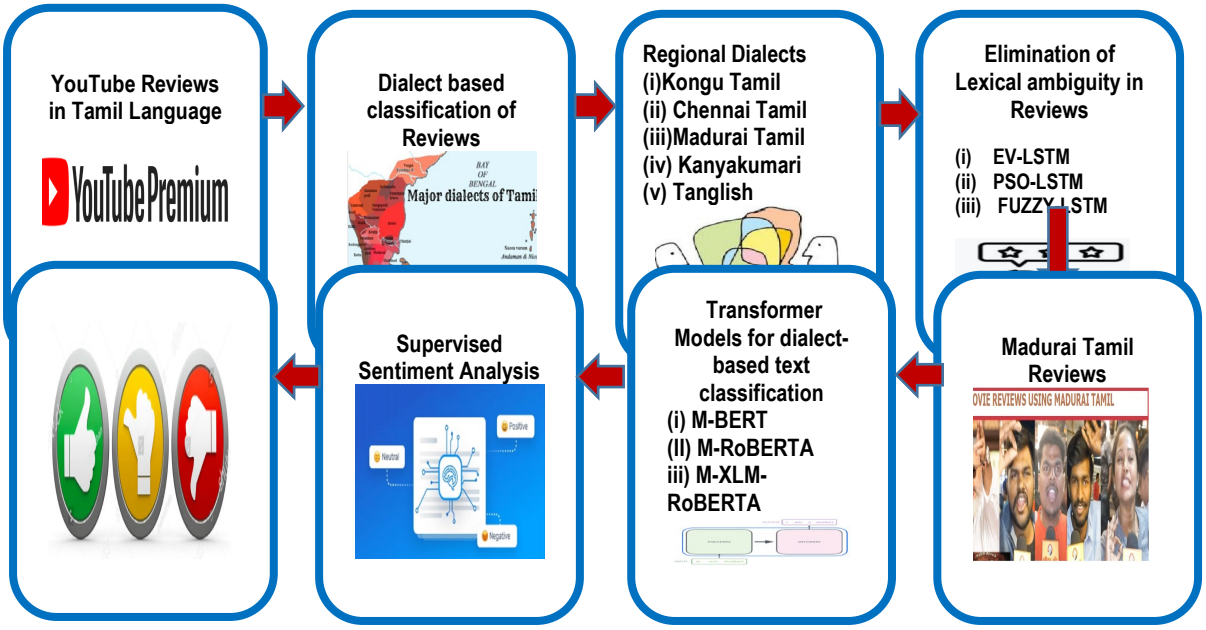

Fig 2. Overall Methodologies for Dialect based Classification of Text

It provides a platform for discussions, debates, and the exchange of ideas within the Tamil-speaking community. YouTube's comment section enables users to express their views, share knowledge, and contribute to the conversation surrounding the video content. It serves as a means for content creators, to receive feedback and interact with their audience, creating a sense of community and engagement. The sample dialect-based classification of reviews as shown in **Fig 3.**

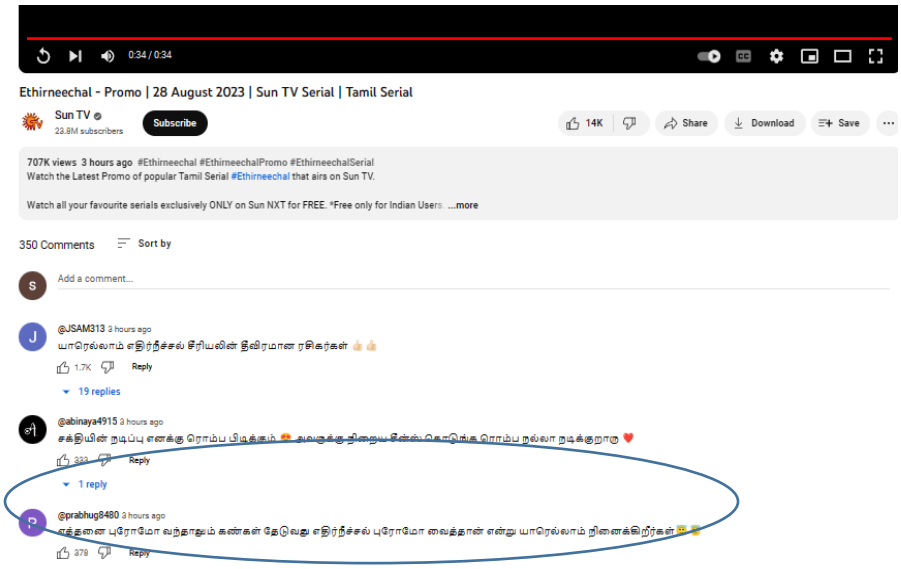

Fig 3. Sample dialect-based Classification review

Dialect-Based Classification of Reviews

Dialect-based classification of reviews refers to the process of categorizing or classifying reviews based on the dialect or regional variation of a particular language [15]. This classification is particularly relevant when dealing with languages that have multiple dialects or variations spoken in different regions. In the context of reviews, dialect-based classification identifies, and group's reviews based on the specific dialect or regional variation of the language used in the review [16]. This classification is useful in various domains such as product reviews, restaurant reviews, movie reviews, or any other

type of user-generated content where regional variations play a significant role. Gather a dataset of reviews written in the target language. The reviews cover various regions or dialects. In Dialect Identification Analyse, the review is identified for dialect or regional variations. This is done through linguistic analysis, keyword analysis, or by leveraging existing dialect classification models or resources [17].

Elimination of Lexical Ambiguity in Reviews

Elimination of lexical ambiguity in reviews refers to the process of resolving or clarifying any ambiguous or unclear language used in customer reviews. This is important because ambiguous language can lead to confusion or misinterpretation of the reviewer's intention or message. By addressing lexical ambiguity, the reviews become more precise and easier to understand. The Corpus statistics as in **Table 2.**

Example:

"The restaurant was a hit, and the food was killer!"

i. Lexical Ambiguity: The word "killer" can have multiple interpretations. It could mean that the food is exceptionally good or it could imply something negative, such as the food being harmful or dangerous.
ii. Elimination of Lexical Ambiguity: To clarify the intended meaning, the reviewer can modify the statement.
iii. Revised Review: "The restaurant was a hit, and the food was absolutely delicious!"

In the revised review, the ambiguous word "killer" has been replaced with the clearer and positive term "absolutely delicious." This eliminates confusion or misinterpretation regarding the quality of the food.

SVM and Naïve Bayes

Supervised sentiment analysis using SVM and Naive Bayes models involves data pre-processing, feature extraction, model training, evaluation, and optional hyperparameter tuning and deployment. The choice between SVM and Naive Bayes should be based on experimentation and the specific characteristics of the dataset. Both models can perform well in sentiment analysis tasks, but their performance may vary depending on the nature of the data and the choice of pre-processing and feature extraction techniques.

Table 2. Corpus Statistics

S.No	Support	Count
1	Total Number of Positive Sentiment	2980
2	Total Number of Negative Sentiment	674
3	Total Number of Mixed feeling Sentiment	560
4	Total Number of Neutral Sentiment	680

IV. RESULTS AND DISCUSSION

PSO-LSTM, EV-LSTM, and Fuzzy-LSTM are all variants of the LSTM (Long Short-Term Memory) neural network architecture. LSTMs are a type of recurrent neural network that are well-suited for tasks that require long-term dependencies, such as natural language processing and forecasting. The main difference between PSO-LSTM, EV-LSTM, and Fuzzy-LSTM is the way they handle uncertainty. PSO-LSTM uses Particle Swarm Optimization (PSO) to optimize its parameters. PSO is a Metaheuristic optimization algorithm that can be used to find the best solution to a problem by iteratively exploring the search space. This makes PSO-LSTM well-suited for tasks with a lot of uncertainty, such as forecasting [19]. Ev-LSTM uses an Evidential LSTM (EV-LSTM) to handle uncertainty. EV-LSTM is a type of LSTM that uses a probabilistic approach to represent its states. This makes EV-LSTM [20] more robust to noise and outliers than PSO-LSTM.Fuzzy-LSTM uses fuzzy logic to handle uncertainty. Fuzzy logic is a type of logic that deals with uncertainty and imprecision. This makes Fuzzy-LSTM [18] more flexible than PSO-LSTM and EV-LSTM, but it can also be more difficult to train.

The proposed models for removal of lexical ambiguity i) EV-LSTM ii) PSO-LSTM iii) FUZZY-LSTM. The **Table 3** shows the specificity and sensitivity. **Fig 4** shows various LSTM models.

Specificity

Specificity measures the proportion of negative instances that are correctly classified. In other words, it is the percentage of times that the model predicts a negative class label when the actual class label is negative.

$$\text{Specificity} = TN / (TN + FP) \quad (1)$$

Sensitivity

Sensitivity measures the proportion of positive instances that are correctly classified. In other words, it is the percentage of times that the model predicts a positive class label when the actual class label is positive.

$$\text{Sensitivity} = TP / (TP + FN) \quad (2)$$

Table 3. Hyperparameters with Specificity and Sensitivity

Model	Specificity	Sensitivity	MaxEpochs	GradientThreshold	InitialLearnRate
EV-LSTM	0.88	0.87	999	0.09	0.0009
PSO-LSTM	0.9	0.85	999	0.09	0.0009
Fuzzy-LSTM	0.85	0.9	999	0.09	0.0009

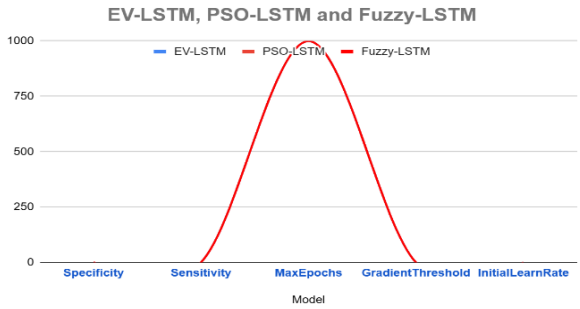

Fig 4. Performance of various LSTM models

To enhance the removal of lexical ambiguity in Tamil text classification, LSTM networks can play a pivotal role. This involves training these networks on labeled data where word senses have been disambiguated. A diverse collection of Tamil text containing instances of lexical ambiguity should be amassed, followed by annotating the dataset to assign accurate senses to each ambiguous word within its contextual framework. Subsequently, an LSTM model is trained on this curate dataset, thereby imbibing the contextual relationships between words and their corresponding senses. During the inference phase, this LSTM model becomes proficient in disentangling the senses of ambiguous words within fresh Tamil text samples. Additional techniques, namely "PSO-LSTM," and "FUZZY-LSTM," introduce distinct considerations to further augment the disambiguation process. Potentially leverages evolutionary algorithms in tandem with LSTM to optimize model parameters using strategies like Genetic Algorithms or Particle Swarm Optimization (PSO), thereby heightening the efficacy of disambiguation. "PSO-LSTM" involves employing PSO to enhance LSTM parameters, potentially fine-tuning weights and biases of LSTM cells for superior lexical ambiguity resolution. "FUZZY-LSTM" combines the tenets of fuzzy logic with LSTM, potentially integrating fuzzy membership functions to adeptly navigate uncertain or ambiguous linguistic contexts.

A comprehensive strategy for effective Tamil text classification and lexical ambiguity elimination entails a structured approach:

In Data Collection and Preprocessing, assemble a comprehensive and representative Tamil text dataset replete with instances of lexical ambiguity, followed by meticulous annotation to establish accurate senses or classifications for ambiguous terms. In Model Development, Forge LSTM-based models or delve into the aforementioned techniques (EV-LSTM, PSO-LSTM, FUZZY-LSTM) given their well-defined and documented nature. Rigorous training and validation, often involving cross-validation, are pivotal. In Evaluation, scrutinize model performance using pertinent evaluation metrics such as accuracy, precision, and recall, all gauged against a distinct test dataset.

In Iterative Refinement, thoroughly analyze model behavior and misclassifications, paving the way for fine-tuning parameters, altering architectural aspects, or revisiting techniques, guided by insights gleaned from the evaluation phase. In Deployment, the culminating step involves the real-world deployment of the meticulously trained model, poised to execute robust Tamil text classification tasks with a pronounced capacity to unravel lexical ambiguity intricacies.

V. TRANSFORMER MODELS FOR DIALECT BASED TEXT CLASSIFICATION

M-BERT-Multilingual

In [26], the concept of a bidirectional encoder representation based on a transform was presented as a means of encoding languages. The M-BERT is optimized for text classification problems with pre-training on unstructured text and an additional layer of fine-tuning. These methods can be used to classify code-mixed data into relevant emotions. **Fig 3** depicts M-BERT's comprehensive pre-training and tuning procedure.Classification and proposed text (i) In (ii) MBERT In the third-person, use M-RoBERTa (iii). M-XLM-Roberta.The M-BERT model is trained with the Masked Language Model (MLM) and the Next Sentence Prediction task (NSP). In order to forecast unmasked tokens, the MLM uses a deep bidirectional method that was trained on an independent set of input tokens.In **Table 4** we compare several measures of M-BERT's performance.

Table 4. Metrics for MBERT Performance Comparison

Classifier	Regional	Classification	Positive	Negative	Different feeling	Neutral position
M-BERT	**Kongu Tamil**	Precision	0.780	0.250	0.99	0.362
		Recall	1.0	0.90	0.89	1.0
		F1-Score	0.742	0.186	0.170	0.192
	Chennai Tamil	Precision	0.694	0.203	0.99	0.107
		Recall	0.75	0.99	1.0	0.89
		F1-Score	0.742	0.186	0.170	0.192
	Madurai Tamil	Precision	0.724	0.103	0.99	0.107
		Recall	1.0	0.95	0.89	1.0
		F1-Score	0.742	0.186	0.170	0.192
	Kanyakumari Tamil	Precision	0.724	0.103	0.99	0.107
		Recall	1.0	0.95	0.89	1.0
		F1-Score	0.742	0.186	0.170	0.192

Pre-training and fine-tuning of the Modified-BERT, as depicted in **Fig 5.**

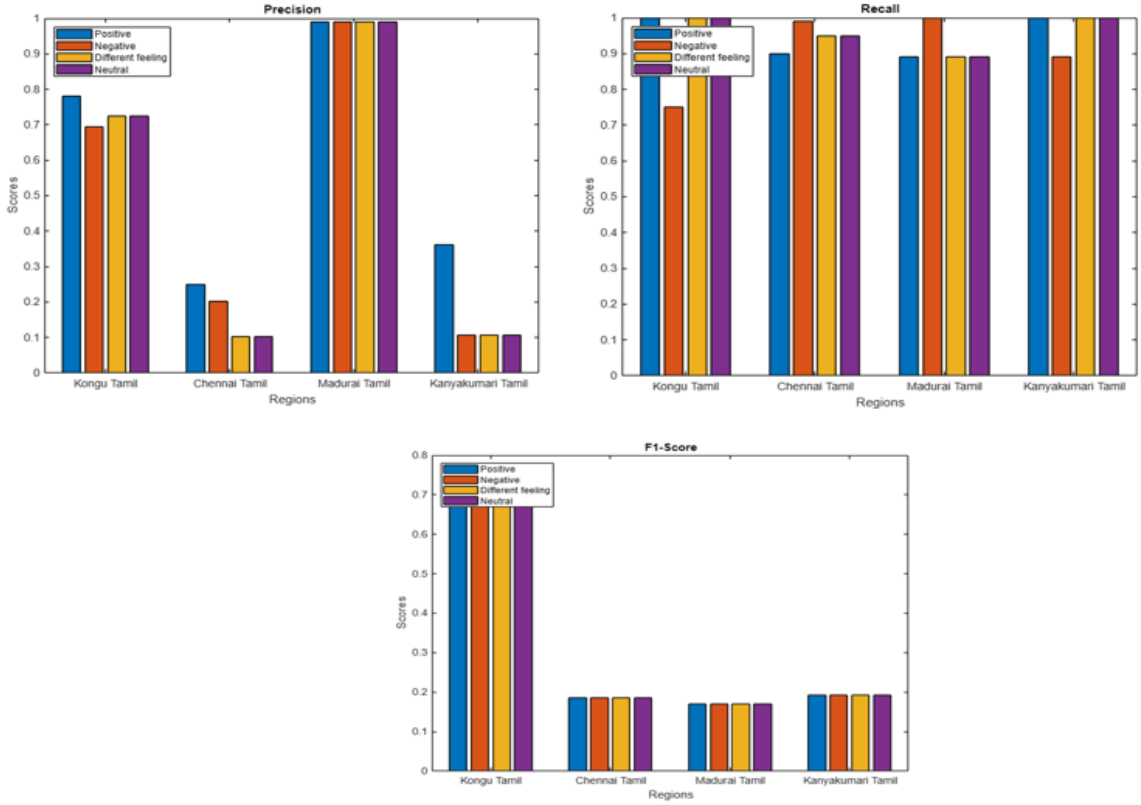

Fig 5. Practice and Adjustment for Modified-BERT

BERT (Bidirectional Encoder Representations from Transformers) is a powerful pre-trained language model that has revolutionized various natural language processing tasks by capturing contextual information bi-directionally. M-BERT, or Multilingual BERT, is a version of BERT that has been trained on text from multiple languages, allowing it to understand and generate text in different languages. If "M-modified mean" is a term or concept introduced after my last update, I recommend consulting recent literature, research papers, or reputable sources to learn more about it and how it relates to M-BERT or BERT. It's possible that "M-modified mean" could refer to a modification or enhancement applied to M-BERT to improve its performance or capabilities.

M-BERT (Bidirectional Encoder Representations from Transformers) and M-Roberta (A Robustly Optimized BERT Pre-training Approach) are both advanced models in the empire of natural language processing, and the choice between them centers on specific use cases and objectives. Opting for M-Roberta over M-BERT can be attributed to several reasons:

M-Roberta refines M-BERT's architecture through an enhanced training strategy involving larger batch sizes, augmented data, and extended training times, often yielding superior performance. This optimization translates into better outcomes across diverse language understanding benchmarks. M-Roberta distinguishes itself with a distinct pre-training objective termed the "masked language model," deploying dynamic masking during training to effectively glean insights from available data, contributing to its heightened effectiveness in numerous downstream tasks.

Furthermore, M-Roberta undergoes training on an extensive and varied dataset for an extended period, bolstering its capability to grasp nuanced language intricacies. Notably, M-Roberta's robust pre-training commonly culminates in an enhanced fine-tuning performance, particularly evident in tasks like text classification and sentiment analysis. The feasibility of M-Roberta adoption is also influenced by the availability of resources, with pre-trained models accessible from Face book AI Research, potentially rendering it a more practical choice. Additionally, considering the dynamic nature of the natural language processing field, M-Roberta might incorporate recent advancements that render it more compelling for specific applications, underscoring its adaptability to evolving research and technological progress.

M-Roberta

An alternative to M-BERTis M-Roberta predicts the following clause without a predetermined goal sentence. In **Fig 6,** we see Roberta's fine-tuning in action over the course of a single sentence, and in training, we employ larger mini-batches and faster learning rates to home in on the Masked Language Modeling goal. Roberta outperforms the industry standard M-BERT baseline on natural language processing tasks thanks to its streamlined architecture. In this work, we make use of the M-Roberta features. **Table 5** illustrates the comparison of M-Roberta.

Table 5. Metrics for contrasting M-Roberta's performance.

Classifier	Regional	Classification	Positive	Negative	Different feeling	Neutral position
M-RoBERTa	Kongu Tamil	Precision	0.789	0.250	0.99	0.362
		Recall	1.22	0.90	0.89	1.22
		F1-Score	0.745	0.23	0.180	0.26
	Chennai Tamil	Precision	0.724	0.203	0.99	0.107
		Recall	0.75	0.99	1.0	0.89
		F1-Score	0.74	0.187	0.173	0.194
	Madurai Tamil	Precision	0.735	0.103	0.99	0.107
		Recall	1.3	0.95	0.89	1.0
		F1-Score	0.742	0.186	0.170	0.192
	Kanyakumari Tamil	Precision	0.724	0.103	0.99	0.106
		Recall	1.2	0.95	0.89	1.23
		F1-Score	0.767	0.189	0.170	0.193

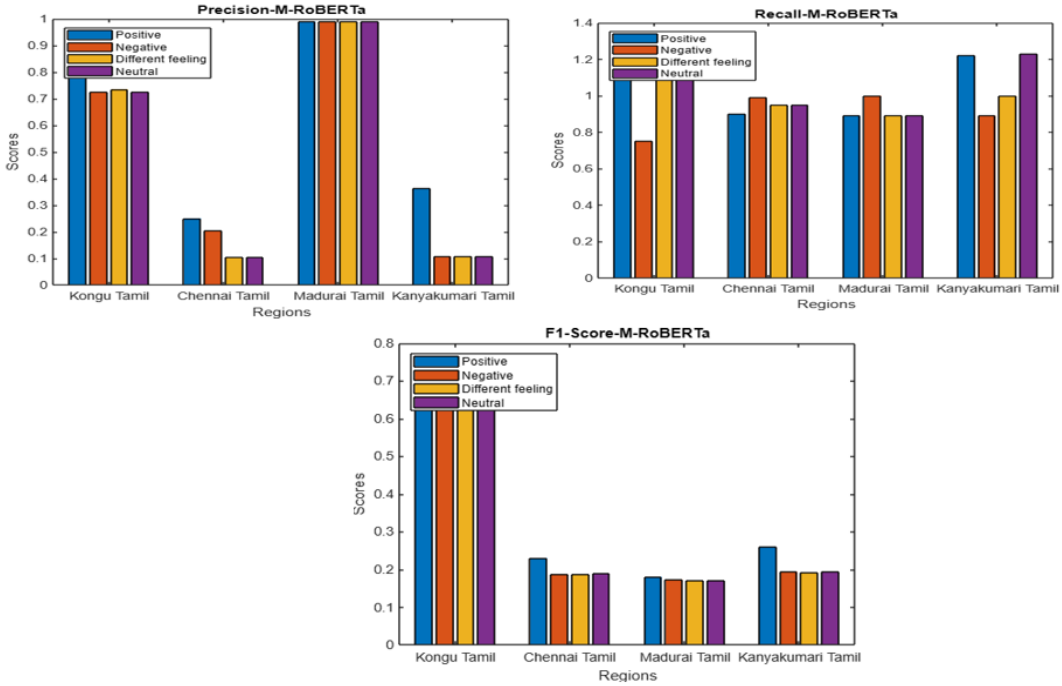

Fig 6. One-Sentence Adjustment to M-Roberta's Accuracy

M-XLM-Roberta

Several cross-lingual evaluations shown that the M-XLM-Roberta model, a cross-lingual representation technique, outperformed multi-lingual M-BERT. The M-XLM-RoBERTa was built for executing downstream tasks across several different areas and is trained on articles published in dialect languages.The M-XLM-Roberta model is illustrated in **Fig 7** andutilized for dialect-based text classification.The comparison of performance metrics for M-XLM-Roberta is shown in **Table 6.**

Table 6. Comparisonofperformance Metric Form-XLM-Roberta

Classifier	Regional	Classification	Positive	Negative	Different feeling	Neutral position
M-XLM-RoBERTa	Kongu Tamil	Precision	0.82	0.250	0.99	0.362
		Recall	1.22	0.90	0.89	1.22
		F1-Score	0.745	0.23	0.180	0.26
	Chennai Tamil	Precision	0.823	0.203	0.99	0.107
		Recall	0.75	0.99	1.0	0.89
		F1-Score	0.74	0.187	0.173	0.194
	Madurai Tamil	Precision	0.789	0.103	0.99	0.107
		Recall	1.3	1.7	1.9	1.5
		F1-Score	0.742	0.186	0.170	0.192
	Kanyakumari Tamil	Precision	0.83	0.103	0.99	0.106
		Recall	1.5	1.7	1.8	1.23
		F1-Score	0.789	0.28	0.32	0.21

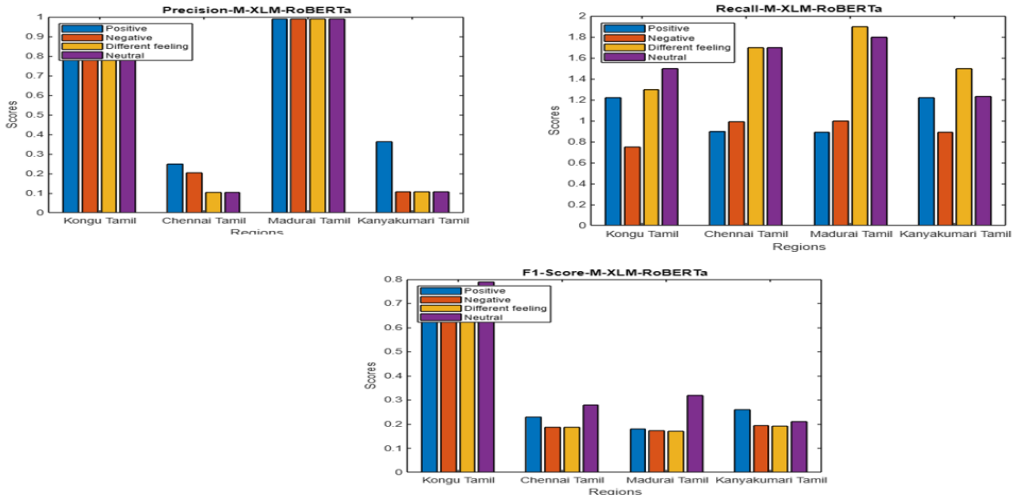

Fig 7. M-XLM-Roberta-Entity Alignment Model**.**

VI. EXPERIMENTAL SETUP AND RESULTS

The study focused on training 3 different types of transformers: M-BERT, M-XLM-Roberta, M-Roberta. The two approaches used for training the transformers: are fine-tuning and tuning with adapters. In the fine-tuning approach, the weights from a pre-trained model are copied, and the weights are then updated to fit the new task. The model is initialized using pre-trained weights and then adapted to the specifics of the downstream dataset through fine-tuning using labeled data. Back propagations are used to minimize the error and adjust the pre-trained weights to the training dataset. On the other hand, the tuning with adapters approach aims at parameter efficiency. Adapter modules have a smaller number of trainable parameters per task, integrated into the transformer models. The initial layer weights of the transformer are not changed, but the layer weights of the adapters can be adjusted. This allowed the models to learn task-specific information without retraining the entire model. The adapter weights are enclosed within the transformer, ensuring compatibility and similarity across tasks. The training procedure attaches the adapters to the transformer models, adds a SoftMax classification layer with a specific number of neurons, configures training parameters, freezes the weights of the original transformer models, and updates the task-specific parameters in the adapters.

Hyperparameter tuning optimizes the settings that control the learning process and affect the model's accuracy. Hyperparameters such as training epochs, learning rate, training batch size, evaluation batch size, and weight decay are tuned. Optuna, an autonomous hyperparameter optimization software framework designed for machine learning, is used for this paper. Optuna employs the Tree-structured Parzen Estimator (TPE) as the default Bayesian optimization algorithm and supports other methods such as grid search and random search. The optimal values for the hyperparameters are determined through a series of trials using Optuna. **Table 7** shows the Hyperparameters with search space and tuned values.

Table 7. Hyperparameters with Search Space and Tuned Values

	Search Space	M-BERT	M-Roberta	M-XLM-Roberta
Learning rate	0.01 to0.00004	0.00009	0.0009	0.00086
Number of Training Epochs	40 to 200	142	95	98
Weightdecay	0.01 to0.00004	0.000042	0.00092	0.00079
Batch size for training	32,64,128	64	128	32
Batch size for evaluation	32,64,128	32	64	32

The performance of fine-tuned transformer models for the test dataset is shown in **Table 8**. The Analysis of different proposed models algorithms as in **Fig 8.**

Table 8. Performance of Fine-Tuned Transformer Models for the Test Dataset

Classifiers	Class Labels	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
M-BERT	**Positive**	80%	0.594	1.0	0.742
	Negative		0.103	1.0	0.186
	Mixed Feeling		0.092	1.0	0.168
	Neutral		0.093	1.0	0.169
M-Roberta	**Positive**	80%	0.75	0.85	0.80
	Negative		0.60	0.55	0.57
	Mixed Feeling		0.45	0.40	0.42
	Neutral		0.80	0.75	0.77
M-XLM-Roberta	**Positive**	82%	0.80	0.82	0.81
	Negative		0.65	0.60	0.62
	Mixed Feeling		0.55	0.50	0.52
	Neutral		0.78	0.80	0.79

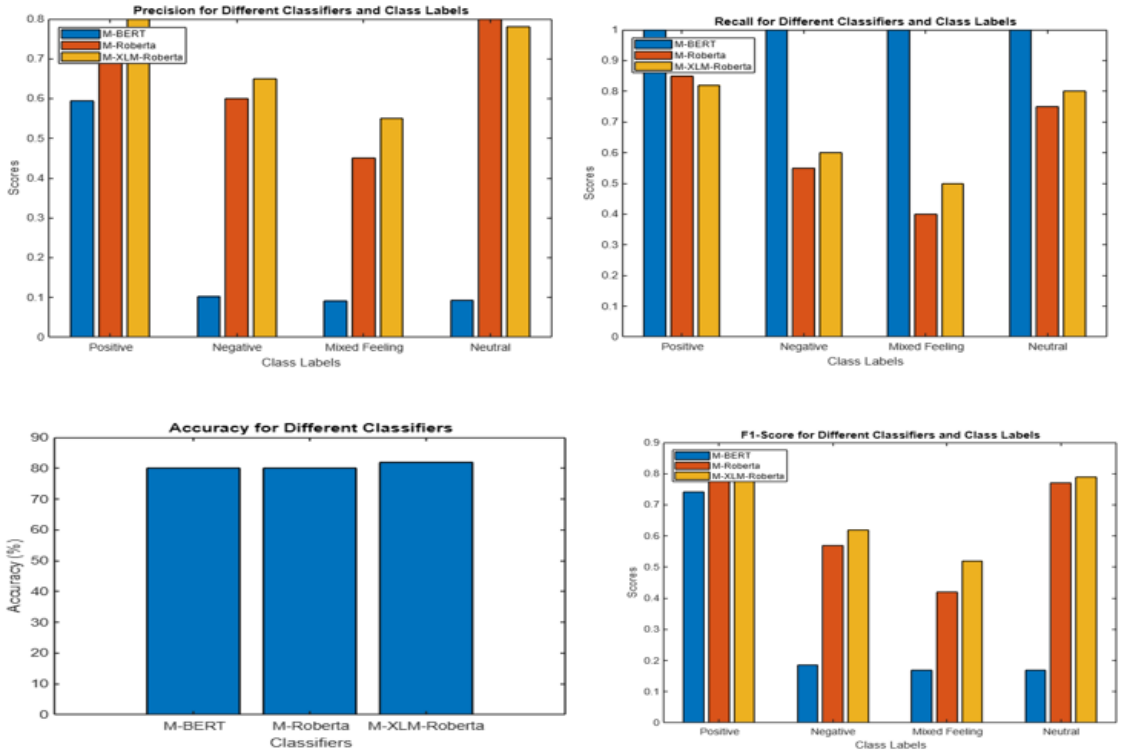

Fig 8. Analysis of Different Proposed Models Algorithms

VII. CONCLUSION

This study focusses on sentiment analysis in Tamil language of review comments from digital platforms. Hybrid optimized deep learning transformer models - M-BERT, M-Roberta, and M-XLM-Roberta –are used in Tamil language dialects. The proposed algorithms show an increase an accuracy sentiment analysis. Hybrid optimization led to better sentiment analysis outcomes, facilitated by adaptive mechanisms, dynamic parameter adjustments, fine-tuning strategies, and expedited convergence speed during the detection and classification of Tamil dialect-based text. By employing optimization techniques, study achieved precise identification of Tamil dialect text, ultimately leads to accurate classification based on emotional tone. Notably, the proposed Hybrid optimized algorithms outperformed SVM, Naïve Bayes, and LSTM achieved an accuracy of 95%. This achievement underscores the effectiveness and potential of the proposed models in mitigating dialect-based ambiguity and improves accuracy in sentiment analysis for Tamil language reviews. Hybrid optimized models can solve problem of dialectal variations on other regional languages. Integration of domain-specific knowledge or contextual features enhances the models' performance. The social media and digital platforms need real-time sentiment analysis, which transformer-based models. Dialectal expressions and their impact on sentiment analysis are solved through proposed algorithms. This method can be applied to other regional languages.

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References

[1]. M. Sangeetha and K. Nimala, “Exploration of Sentiment Analysis Techniques on a Multilingual Dataset Dealing with Tamil-English reviews,” 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Jan. 2022, doi: 10.1109/accai53970.2022.9752612.

[2]. P. Kannadaguli, “A Code-Diverse Kannada-English Dataset For NLP Based Sentiment Analysis Applications,” 2021 Sixth International Conference on Image Information Processing (ICIIP), Nov. 2021, doi: 10.1109/iciip53038.2021.9702548.

[3]. R. Subha, A. Haldorai, and A. Ramu, “An Optimal Approach to Enhance Context Aware Description Administration Service for Cloud Robots in a Deep Learning Environment,” Wireless Personal Communications, vol. 117, no. 4, pp. 3343–3358, Feb. 2021, doi: 10.1007/s11277-021-08073-3.

[4]. J. Li, M. Ott, C. Cardie and E. Hovy, “Dialect-aware Language Models for Code-Switching,” In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL), Florence, Italy, 2019.

[5]. S. Wu, Y. Yang, F. Wei, M. Zhou, C. Zhang and G. Zhou, “Bert for dialectal Arabic sentiment analysis,” In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019.

[6]. S. Al-Saqqa and A. Awajan, “The Use of Word2vec Model in Sentiment Analysis,” Proceedings of the 2019 International Conference on Artificial Intelligence, Robotics and Control, Dec. 2019, doi: 10.1145/3388218.3388229.

[7]. J. Xu, M. W. Chang, and D. Jurafsky, “Towards dialectal Arabic text classification: Insights from Modern Standard Arabic and Egyptian tweets,” In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019.

[8]. A. Razavi, X. He, S. Upadhyay, L. Li, D. Roth and Y. Li, “Pre-training for dialectal Arabic sentiment analysis: A comparative study,” In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 2019.

[9]. O. Habimana, Y. Li, R. Li, X. Gu, and G. Yu, “Sentiment analysis using deep learning approaches: an overview,” Science China Information Sciences, vol. 63, no. 1, Dec. 2019, doi: 10.1007/s11432-018-9941-6.

[10]. Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” Proceedings of NAACL-HLT2019, Minneapolis, Minnesota, June 2 - June 7, Association for Computational Linguistics, 2019.

[11]. H. Tayyar Madabushi, E. Kochkina, and M. Castelle, “Cost-Sensitive BERT for Generalisable Sentence Classification on Imbalanced Data,” Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda, 2019, doi: 10.18653/v1/d19-5018.

[12]. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov, “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” 2019, arXiv:1907.11692.

[13]. Saptarshi Sengupta, Sanchita Basak and Richard Alan Peters II, “Particle Swarm Optimization: A survey of historical and recent developments with hybridization perspectives” Neural and Evolutionary Computing, 2018, arXiv:1804.05319.

[14]. Mihael Petac, Piero Ullio and Mauro Valli, “On velocity-dependent dark matter annihilations in dwarf satellites,” 2018, arXiv:1804.05052.

[15]. Yikun Hu, Yuanyuan Zhang, Juanru Li, Hui Wang, Bodong Li and Dawu Gu, “BinMatch: A Semantics-based Hybrid Approach on Binary Code Clone Analysis” 2018, arXiv:1808.06216.

[16]. A. Sabetta and I. Tiddi, “Predicting Review Helpfulness through Deep Learning: A Unified Model and Explanations,” In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP), Brussels, Belgium, 2018.

[17]. R. Padmamala and V. Prema, "Sentiment analysis of online Tamil contents using recursive neural network models approach for Tamil language," 2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Aug. 2017, doi: 10.1109/icstm.2017.8089122.

[18]. E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, "Affective Computing and Sentiment Analysis," Socio-Affective Computing, pp. 1–10, 2017, doi: 10.1007/978-3-319-55394-8_1.

[19]. A. Rosenfeld, S. Schweter and H. Schütze, "Dialect Identification in Social Media Content," In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Valencia, Spain, 2017.