JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

CLIENT EVALUATOR USING DIFFERENT LANGUAGES

[1]Dr. K.M. PONNMOLI, [2]Dr. P SANKARAN
[1]TGT,[2]Assistant professor
[1]Department of Computer Science,[2]Department of Commerce,
[1]AAGASC, Karaikal- 609 602,[2]KMKGIPGS&R, Karaikal-609602
[1]kmponnmoli@yahoo.in, [2]psankaran12@gmail.com

Abstract :Customer feedback is considered one of the most significant assets for a business. It provides valuable insights into product performance, user satisfaction, and areas that need improvement. However, the continuous analysis of a large volume of unstructured feedback in various languages can be quite laborious and inefficient. This paper presents a multilingual approach to automatically categorize reviews in English, Hindi, and Gujarati into categories such as bug reports, feature requests, positive feedback, and others. The proposed system utilizes MuRIL (Multilingual Representations for Indian Languages), a transformer-based model specifically designed for Indic languages. This model effectively captures contextual meanings and cross-lingual subtleties. When fine-tuned on a well-balanced multilingual dataset, it shows strong performance in classifying different language structures. This framework is scalable, automated, and functions in a cloud environment for feedback analysis. It aids organizations in making quicker, data-driven decisions, improving product quality, and enhancing overall customer satisfaction**.**

***IndexTerms* - Multilingual NLP, MuRIL, Customer Feedback Classification, Indic Languages, Transformer Models, Cross-Lingual Learning, Sentiment Analysis, AI-driven Analytics.**

I. INTRODUCTION

In the modern technological age of digital transformation, there is a growing dependence on customer feedback to assess the effectiveness of a product and the efficiency of a system. With the significant increase in various online tools such as mobile applications, e-commerce platforms, surveys, and social media channels, the growth in customer feedback has been substantial. Analyzing large amounts of such unstructured data is unsustainable and highly challenging, indicating the necessity for intelligent analysis of customer feedback. To address the challenges, this paper suggests the design of a system called Multilingual Customer Feedback Analyzer, which will automatically classify the customer feedback, which will be in English, Hindi, and Gujarati, and map it to different categories, which may be bug reports, feature requests, positive feedback, and other categories. For this purpose, the presented system will utilize the MuRIL, which stands for Multilingual Representations for Indian Languages. It is based on the transformer model and has been specifically designed to learn contextual representations for Indian languages.

To ensure scalability as well as accessibility, the system is set up with a cloud platform like Google Colab for efficient training of the models. The proposed pipeline helps to save time, automate more processes, speed up the iteration of the product, as well as enable organizations to effectively leverage insights from multilingual feedback.

1.1. Motivation

1) Manual Analysis Becomes Impractical at Scale: Currently, organizations are inundated with numerous customer reviews, which are processed through platforms such as mobile applications, websites, and social media. The task of processing these reviews is not only labor-intensive but also lacks consistency. Additionally, managing different languages presents significant challenges, as it is nearly unfeasible.

2) Regional Language Feedback Often Gets Ignored: In nations that provide services in multiple languages, like India, a majority of users tend to express their opinions in Hindi or Gujarati instead of English. Nevertheless, most existing feedback analysis methods are designed primarily for the English language. Consequently, insights derived from regional languages often remain overlooked.

3) Basic Tools Fail to Capture Meaning Correctly: Typically, keyword-based systems struggle to accurately interpret user feedback, as they lack contextual understanding. For example, when the system identifies the word "bug," it may incorrectly categorize this as negative feedback, which is not always the case.

4) Language Variations Create Additional Challenges: Hindi and Gujarati exhibit considerable morphological richness, flexible syntax, and colloquial expressions. Furthermore, reviewers may switch between formal and informal tones in their feedback.

5) Making a Scalable and Practical Solution: As companies grow, customer feedback comes in at a very large number. It has become unthinkable for someone to manually read and categorize thousands of reviews amid their varied languages. The business requires a system that can efficiently handle this volume without slowing down the process. A scalable automated solution for review processing will speedily have the team address the issues and improve the products

1.2. Proposed Methodology

1) Data Collection: Customer feedback is gathered through multiple channels, including surveys, support tickets, app store reviews, and social media. Since users express their opinions in various languages, the dataset will include feedback in English, Hindi, and Gujarati to ensure a realistic multilingual representation.

2) Data Preprocessing: The raw text feedback undergoes processing and formatting before being input into the model through Tokenization, Lemmatization, Stop Word Removal, and Language Processing.

3) Text Embedding using MuRIL: This technique is employed to convert the cleaned text into contextual vectors. Unlike traditional embeddings, MuRIL captures deep contextual meanings across multiple Indian languages, enabling effective cross-lingual interpretation.

4) Strategies used in learning: Supervised Learning Approach: The MuRIL model is trained on labeled feedback data to classify reviews into categories such as bug reports, positive feedback, feature requests, and others. This process allows the model to recognize significant patterns and linguistic variations across different languages.

5) Unsupervised Learning Approach: Alongside classification, clustering techniques like K-Means and DBSCAN are utilized to uncover hidden themes and emerging patterns in newly received feedback. This approach aids in identifying new issues that may not fit into predefined categories.

6) Model Evaluation: The performance of the classification model is evaluated by using standard evaluation measures such as accuracy, precision, recall, and F1-score. These metrics are used to make sure that the model is performing the same across all supported languages.

7) Cloud Based Deployment: The system is designed to be used in cloud environment like on Google Colab, supporting scalable training and inference without special hardware. This configuration makes the solution practical and accessible for use by the general public.

1.3. Problem Statement

In the current knowledge-driven economy, obtaining customer feedback has become increasingly vital for organizations aiming to improve their products and services. The emergence of digital platforms has led to a surge of users sharing their thoughts on your products, experiences with customer service, and the performance of employees through mobile applications, websites, support portals, emails, and social media. Although this feedback holds significant value, it is often unstructured and conveyed in natural language.

The situation is further complicated in multilingual nations like India, where users express their opinions not only in English but also in Hindi, Gujarati, and other languages. Most traditional feedback analysis systems are either tailored exclusively for English or rely on simplistic keyword-based methods. Consequently, feedback written in regional languages is frequently overlooked, misclassified, or not fully leveraged.

The manual processing of such multilingual feedback is not only labor-intensive but also inconsistent and prone to human error. As the volume of data increases, manual approaches become impractical and unscalable. This, in turn, adversely affects response times, delays product enhancements, and potentially leads to a decline in overall customer satisfaction.

Current market tools are often either too generic or overly specialized for their respective domains, failing to accurately grasp the context, tone, and intent of the feedback. Numerous systems struggle to differentiate between various types of feedback, such as bug reports, feature requests, critical issues, or general opinions. Furthermore, a single review may encompass multiple sentiments or intentions, complicating classification even further. Traditional rule-based systems lack the flexibility required to manage such complexity.

Another shortcoming of the existing methods is that they are hard to be generalized by domain-specific vocabulary and linguistic variation in multiple languages. Moreover, many of the existing approaches do not make an explicit use of either real-time processing or cloud-based scalability, thus limiting their usefulness in such an ever evolving and fast world.

Recent advances in NLP, especially transformers and transformer-based approaches like MuRIL (Multilingual Representations for Indian Languages), have unlocked new opportunities to deal with mixed text more intelligently. MuRIL is trained on Indic languages and hence can understand contextual substance irrespective of the structure of language.

Thus, there exists an urgent requirement for intelligent and scalable multilingual feedback analysis systems which can accurately analyze and classify customers' review comments across languages. The proposed multilingual customer feedback analyzer attempts to bridge this gap by using MuRIL based context-aware embeddings and deep learning methodologies to convert raw, unstructured feedback into structured and operational insights. Instead, a system like this would enable quicker decisions and awareness of what needs attention, along with more responsive customer interaction.

1.4. Challenges

Creating a Customer Feedback Analyzer can present several challenges, including:

1) Unstructured and Noisy Data: The customer feedback can be written in informal language, mixed scripts, slangs, emojis, grammatical errors, etc., particularly in Hindi and Gujarati which makes it difficult to be processed with high accuracy.

2) Ambiguity in Feedback: A single review can involve different intentions in the same review such as positive comments with complaints that makes the category difficult to define.

3) Limited Labeled Multilingual Data: It can take much time and be extremely costly to develop labeled feedback data in several languages for fine-tuning transformer-based model.

4) Contextual Misinterpretation: Even advanced models might face problems in situations of sarcasm, implicit meaning or region-based expressions for uprising incorrect classification.

5) Scalability Constraints: Processing big data of multilingual feedback, almost instantly can be pretty demanding in terms of computational resources.

6) Domain Adaptability: A model trained with feedback from a domain might not be able to generalize well to another without further fine tuning.

7) Multilingual Complexity: Multiple language feedbacks in English, Hindi, and Gujarati make feedback processing more complex because there is a high usage of linguistic variability, script and contextual differences.

8) System Integration: Integrating the analyzer with an existing set up of business dashboards, support systems, or cloud infrastructure is technically challenging.

9) Real-Time Processing Limitations: Real has low limited hardware and test deployment constraints are to be hit in obtaining low-latency multilingual classification.

II. Literature Review

1) Customer Feedback Analysis Overview: The analysis of customer feedback has been extensively researched utilizing machine learning, deep learning, and natural language processing techniques. Researchers have developed various models for sentiment detection, complaint classification, and automated feedback interpretation aimed at improving decision-making and customer relationship management.

2) Emotion-Based Feedback Analysis: Numerous studies have focused on emotion-based feedback analysis, employing machine learning models. These approaches combine computer vision and deep learning techniques, such as Convolutional Neural Networks (CNN), to recognize emotional states and interpret user responses. Some frameworks also incorporate privacy-preserving mechanisms to safeguard sensitive user information. While there are encouraging outcomes regarding automated feedback comprehension with these systems, they often rely on small datasets and require significant computational resources, making real-world implementation challenging due to data variability and infrastructure complexity.

3) Privacy-Saving and Distributed Learning Methods: The analysis of feedback with a focus on privacy has also been explored through distributed learning methods. In these frameworks, model training occurs across multiple data sources simultaneously, ensuring user privacy through secure learning systems. Although these methodologies enhance data security, they introduce challenges related to computational overhead, data heterogeneity, and communication efficiency. Furthermore, many of these systems face obstacles in real-time implementation and lack support for various input modalities.

4) Traditional Machine Learning and Deep Learning Sentiment Analysis: Traditional Customer feedback sentiment analysis approaches have received widespread use. Comparative studies of machine learning algorithms, which include Logistic Regression, Support vector machines, random forest and Naive Bayes have proved to be effective in terms of their ability to determine positive and negative sentiment. Newer neural networks like Long short term memory (LSTM) networks have exhibited better results as they can extract sequential dependencies in text better. Nevertheless, the models usually have a problem with neutral sentiment prediction, need substantial labels, and might be ineffective in terms of inter-domain generalization.

5) Transformer-Based Feedback Classification: The developments in deep learning have further enhanced feedback classification by using Transformer-based architecture. Such models give contextual descriptions of language and have shown better performance than the traditional recurrent neural networks. Transformer-based methods even though correct are usually memory-intensive and demand domain adaptation fine-tuning. In addition to that, most of the implementations concentrate more on sentiment polarity instead of detailed feedback categorization.

6) Complaint Classification with Neural Networks: Consumer complaint classification is another example that has been conducted through deep neural network designs and word encoding methods. Models like LSTM, BiLSTM, CNN and lightweight variants of

transformers have demonstrated good classification performance. The difficulties of unbalanced datasets, however, having little to no contextual expression in the conventional embeddings, and not being able to process information in real-time still affect the system reliability.

7) NLP-Based Automated Classification Systems: There has been research into automated complaint classification based on the general NLP preprocessing algorithms and supervised learning algorithm. Although these methods can perform at mediocre levels, they are usually faced with either short or ambiguous text, overlapping semantic categories, and lack of uniform data quality. Such constraints indicate that models are in need to capture the contextual meaning and linguistic variability, better.

8) Shortcomings of Current Solutions:Despite the prevalence of progress in automated feedback analysis before it, there still exist a number of limitations to the solution. Most of the existing systems also concentrate on single-language datasets especially English and do not support multilingual environments. This limits the issue of feedback analysis tools in areas that have a high language diversity. In addition, the majority of works focus on the sentiment classification instead of fine-grained categorization of feedback into actionable business relevant classes.

9) Multilingual Transformer Models Emerging: To overcome these shortcomings recent studies have started to consider multilingual language models based on transformers that can be used to represent contextual associations among languages. These models provide better semantic insight and can allow better classification of radio varied textual inputs. Based on these developments, the current study is on the Way of Multilingual Customer feedback Classification through Context-definite representations designed in a linguistically heterogeneous context. The proposed system will offer flexible and feasible solution to the analysis of feedback in the real world by facilitating various languages and categorization of feedback based on feedback sub-groups.

III.METHODOLOGY

3.1System Architecture

The proposed MCFA uses MuRIL (Mulitlingual Representations for Indian Languages) as core transformer encoder. MCFA, or Multilingual Customer Feedback Analyzer, is a modular NLP pipeline designed to customer feedback in English, Hindi, and Gujarati.

The architecture includes the following main components.

1) Input Section: The CSA file or web form accepts unprocessed text of customer feedback. Unicode Text for Hindi (Devanagari code) and Gujarati script supported.

2) Data Preprocessing: This includes various steps that help ensure the classifier can classify the reviews in a proper manner. This includes steps like,

a) Letter case conversion in text normalization and clean up unwanted spaces and special symbols.

b) Unicode standardization.

c) Detection of language for analysis optional.

3) MuRIL Tokenizer:

a) It first employs the WordPiece tokenization.

b) Turns text into input IDs attention masks and token type IDs.

c) Manages inputs with different scripts.

4) Encoder for Transformer (MuRIL):

a) A transformer encoder with 12 layers.

b) Utilization of self-attention mechanism for contextualized embedding representations.

c) Yields a sentence representation context with a CLS token embedding

5) Classification Component:

a) All-purpose dense layer.

b) Function activation softmax.

c) Gives probability distribution over four classes such as, Bug Report, Feature Request, Positive Feedback and other.

6) Module for Evaluation and Prediction: It provides estimated labels, calculates key metrics, including things like the Accuracy and Precision, and finally saves outcomes for review.

3.2 Data Description

The customer feedback multilingual dataset used in the present work encompassed four predefined classes, such as:
Bug report, Feature request, Positive Feedback, Other.

1) Language Distribution: The dataset includes reviews of various languages such as,

a) English feedback.

b) Reviews in Hindi.

c) Gujarati Reviews (Gujarati Lettering)

Meticulous curation was undertaken of each language dataset to make sure the dataset has accuracy of the script, distinct samples (no repeats), uniform spread across classes.

2) Size of Dataset: To achieve consistency across experiments, we made sure to maintain the same samples per class within the constraints of the data.

3) Data Features: The dataset included data that covered a wide range of features such as,
a) Brief and detailed feedback.

b) Use a friendly tone.

c) Mix of Languages.

d) Complaint patterns of real customers.

3.3 Data Preprocessing

1) Cleaning: This step ensures the model got a clean, well- balanced dataset, this was done by making use of the following,

a) Duplicates removed.

b) Removed null entries.

c) Removed extra spaces.

2) Unicode Management:

a) Encoding (UTF-8).

b) Retained scripts of Hindi and Gujarati.

3) Encoding Labels: The labels were assigned as,

Label	Encoding Value
bug_report	0
feature_request	1
positive_feedback	2
other	3

4) Train-Test Split: The model was trained on 80% of the overall dataset, and for the evaluation the remaining 20% of the dataset was used (test set), this was done to strategically divide to maintain class balance.

3.4 Model picking: MuRIL

MuRIL is a transformer-based Indian language model specifically designed for Indian languages. This is the chosen model for this work.
1) Reason for choosing MuRIL: It is trained on over 17 Indian languages; it also takes care of transliterated and code-mixed text. Indic languages perform better on multilingual BERT models than their generic counterparts. Also, as it is a transformer-based model, the embeddings are context aware and make use of self-attention.

2) Fine-tuning Strategy: For fine-tuning MuRIL to our use-case, we made use of several steps to ensure a robust model. This was done by,

a) Loading Pretrained MuRIL weights.

b) A layer for classification was added.

c) Making use of a detailed dataset to fine-tune the model.

d) Utilizing cross-entropy loss function.

e) AdamW optimizer with learning rate scheduling.

3.5 Training Configuration

The training setup made use of the following,

1) Optimizer: AdamW

2) Loss Function: Cross-Entropy loss

3) Batch size: 16-32

4) Learning Rate: 2e-5 to 5e-5

5) Epochs: 3–5

Along with this, early stopping was implemented to prevent overfitting.

F. Evaluation Metrics

To assess model performance, the following metrics were computed.

1) Accuracy

2) Precision

3) Recall

4) F1-Score (Macro and Weighted)

5) Confusion Matrix

The Macro-F1 was given more priority due to the multi-class classification of our project and the chance of a potential class imbalance.

IV. RESULT

4.1 Model Performance Comparison

Our experimental assessment contrasted the performance of our fine-tuned Muril and Zero-Shot RoBERTa large architectures on the customer feedback classification task.

Table I shows the complete metrics across both models:

Table 4.1: Comparing Muril and RoBERTa Large

Model	Accuracy	Precision	Recall	F-1 Score
Muril (Fine-Tuned)	0.92	0.92	0.92	0.90
RoBERTa Large (zero-shot)	0.36	0.48	0.39	0.31

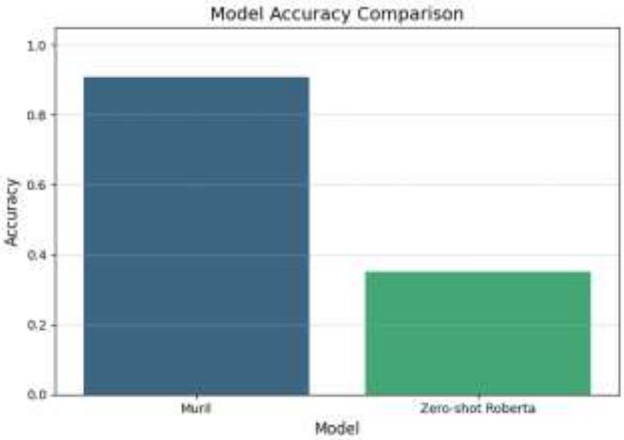

Fig. 4.1 Model Accuracy Comparison

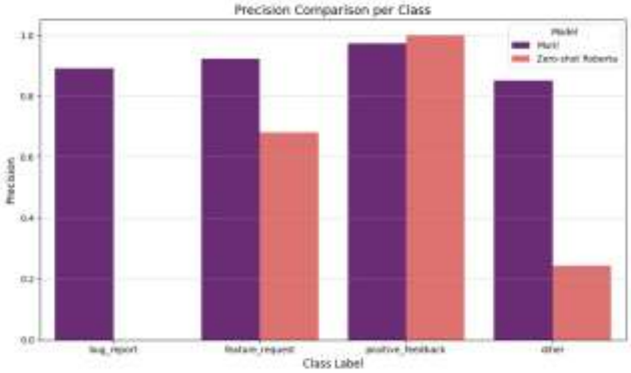

Fig. 4.2 Precision Comparison per Class

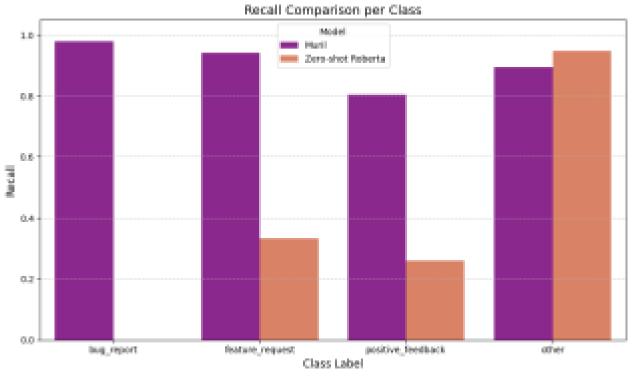

Fig. 4.3 Recall Comparison per Class

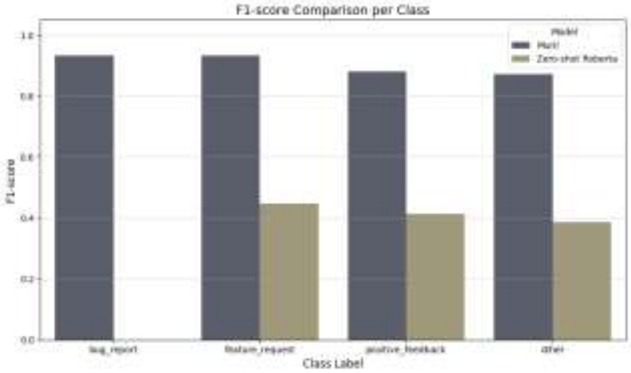

Fig. 4.4 F1-score Comparison per Class

MuRIL model outperforms the zero-shot baseline when evaluated with all metrics. MuRIL completion achieves a 91% overall accuracy. As presented in Table X, the macro-averaged precision, recall and F1-score for MuRIL is about 0.90. It is clear from this result that MuRIL performs consistently across all classes.

The RoBERTa-Large zero-shot model performs poorly, achieving only 35% accuracy and a macro F1 score of 0.31. While the precision looks moderate for some classes, the recall is very imbalanced- it does not get any correct classification for the bug_report class.

Fine-tuning with multilingual task-specific data is much more effective in improving classification robustness and classes discriminate ability than the zero-shot approach. Domain adaptation and multilingual contextual learning is one of the reasons.

4.2 Class-Wise Analysis

To get a better understanding of the model's performance, we calculated the precision, recall, f1-score and the support of both the models, that is the fine-tuned Muril and the zero-shot classification done by RoBERTa large.

The below table shows the results that were achieved by the Fine-Tuned Muril model. The testing dataset consisted of reviews of various languages such as English, Hindi, Gujarati which were not seen by the model during the training phase to ensure a fair evaluation.

Table 4.2: Muril (Fine-Tuned) Class-wise results

Label	Precision	Recall	F1-Score	Support
bug_report	0.89	0.98	0.93	50
feature_request	0.92	0.94	0.93	51
positive_feedback	0.97	0.80	0.88	46
other	0.85	0.89	0.87	38

The below table shows the results that were achieved by the RoBERTa Large model, where the classification was done by zero-shot classification. The same dataset which was used for getting the class-wise results for the fine-tuned Muril was used here too to ensure a fair evaluation.

Table 4.3: *RoBERTa (Zero-Shot) Class-wise results*

Label	Precision	Recall	F1-Score	Support
bug_report	0.00	0.00	0.00	50
feature_request	0.68	0.33	0.45	51
positive_feedback	1.00	0.26	0.41	46
other	0.24	0.95	0.39	38

Key observations from the classification reports:

1) Bug Report: MuRIL model in fine-tuning fashion resonates very well (F1 = 0.93, Recall = 0.98) capturing almost all bug related feedback. However, the zero-shot RoBERTa-Large model does not produce any results (F1 = 0.00, Recall = 0.00). This indicates that the bug descriptions that are often subtle do not work either.

2) Feature Request: MuRIL outperformed zero shot model in this task. While the latter was able to identify some features (F1 = 0.45), its recall (0.33) suggests many were wrongly classified. Overall, MuRIL gave a strong and balanced performance. (F1 = 0.93, Recall = 0.94).

3) Positive Feedback: MuRIL showed that it understood the context well (F1 = 0.88, Recall = 0.80). The zero-shot model was very accurate (1.00) but not very good at remembering (0.26). This means that it only predicted this class when it was very sure, and it missed most of the positive feedback cases.

4) Other: The zero-shot model had an unusually high recall (0.95) for the other category, which suggests that it was very likely to predict this class when it wasn't sure. MuRIL also did well (F1 = 0.87), but its predictions were more evenly spread across categories, which shows that it had better discrimination ability.

4.3 Error Analysis

A manual review of incorrectly labeled samples revealed the following findings:

1) The zero-shot RoBERTa-Large model failed to identify any bug_report instances, resulting in a Recall of 0.00. Instead, it frequently misclassified these instances into the incorrect category, predominantly the other class. This indicates that the model struggles to differentiate between general feedback and descriptions of functional defects unless it adapts to the specific task at hand.

2) The zero-shot model exhibited an exceptionally high recall for the 'other' category (0.95), suggesting that it was inclined to predict this class when uncertain. Numerous feature requests and positive feedback samples were misclassified, adversely affecting overall performance.

3) The low recall (0.26) for positive feedback indicated that the zero-shot model only recognized a limited number of clearly positive instances, overlooking moderately expressed appreciative comments. This highlights the conservative nature of predictions.

4) The fine-tuned MuRIL model demonstrated significantly fewer errors across all categories, achieving balanced recall values ranging from 0.80 to 0.98. Most remaining inaccuracies occurred in ambiguous situations where feature suggestions were conflated with appreciation comments. This implies a minimal degree of semantic overlap rather than an issue with the structural model.

4.4 Discussion

The results provide significant insights into the impact of fine-tuning and multilingual pretraining on this evidence. Additionally, they have broader implications for structured text classification tasks. The zero-shot configuration primarily depends on the semantic similarity between labels and text rather than learning a task in a discriminative manner.

Consequently, while large pretrained language models such as RoBERTa-Large exhibit a robust general semantic understanding, this limitation is particularly significant in multi-class scenarios where subtle class distinctions are learned. An evaluation of the zero-shot version of RoBERTa-Large reveals markedly different behavior compared to the supervised fine-tuned MuRIL. It seems that there is a considerable class imbalance, leading the model to predominantly struggle with predicting one specific class.

As a result, the bug report fails entirely in this context. Therefore, zero-shot inference is unable to identify operational or technical bug descriptions. In these cases, it necessitates contextual and domain-specific cues, rather than relying solely on direct lexical context. In contrast, the fine-tuned MuRIL model exhibited substantial stability at the macro level, along with balanced discrimination across classes. MuRIL was pre-trained on a significantly larger corpus of Indian language text, which enables it to learn superior token-level representations for the Hindi and Gujarati scripts. Through supervised fine-tuning, we guide the model to understand the ideal decision boundaries within the common embedding space to effectively differentiate between the semantic overlap of positive review and feature request classes. Given that both macro precision and recall values are exceptionally high and consistent, we can infer that the model's internal contextual representation remains stable.

V. Conclusion

The research introduces a multilingual framework aimed at classifying customer feedback, alongside an evaluation of a fine-tuned MuRIL model in comparison to a zero-shot RoBERTa-Large baseline for the same classification task. The experimental results indicate that fine-tuning the MuRIL model significantly enhances classification performance on multilingual inputs, including English, Hindi, and Gujarati. The fine-tuned MuRIL achieves an overall accuracy of 91% and a macro f1-score of 0.90, demonstrating strong performance across all categories: bug report, feature request, positive feedback, and others. In contrast, the zero-shot RoBERTa-Large only reaches 35% accuracy with a macro f1-score of 0.31.

A notable class imbalance exists, as the zero-shot RoBERTa-Large classifier predominantly predicts the “other” category. Additionally, it fails to identify any bug reports, highlighting the limitations of general-purpose language models when applied to domain-specific tasks. Error analysis further indicates that the zero-shot approach struggles with class boundary separation, leading to predictions that often default to high-frequency categories.

Conversely, the fine-tuning process allowed the MuRIL model to establish clearer decision boundaries and effectively capture multilingual contextual semantics. These findings underscore the necessity of domain-specific training and the incorporation of multilingual textual context in real-world customer feedback systems. Although zero-shot models offer convenience and can be utilized for rapid sandbox trials, they fall short for sensitive and high-stakes systems that require 1-to-N classification. Looking aheade.

References

[1]S. Khanuja, D. Khandelwal, A. Singh, N. Kunchukuttan, and P. Goyal, “MuRIL: Multilingual Representations for Indian Languages,” *arXiv preprint arXiv:2103.10730*, 2021.
[2]Y. Liu, M. Ott, N. Goyal, et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach,” *arXiv preprint arXiv:1907.11692*, 2019.
[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. NAACL-HLT*, Minneapolis, MN, USA, 2019, pp. 4171–4186.
[4] A. Vaswani et al., “Attention Is All You Need,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 5998–6008.
[5] T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” in *Proc. EMNLP: System Demonstrations*, 2020, pp. 38–45.
[6] A. Conneau et al., “Unsupervised Cross-lingual Representation Learning at Scale,” in *Proc. ACL*, 2020.
[7] R. Pires, E. Schlinger, and D. Garrette, “How Multilingual is Multilingual BERT?” in *Proc. ACL*, 2019, pp. 4996–5001.
[8] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *arXiv preprint arXiv:1301.3781*, 2013.
[9] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[10] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[11] Y. Kim, “Convolutional Neural Networks for Sentence Classification,” in *Proc. EMNLP*, 2014, pp. 1746–1751.