Towards Effective Emotion Analysis in Low-Resource Tamil Texts

Priyatharshan Balachandran[1] **Uthayasanker Thayasivam**[1]
Randil Pushpananda[2] **Ruvan Weerasinghe**[2]
[1]Department of Computer Science & Engineering, University of Moratuwa, Sri Lanka
[2]University of Colombo School of Computing, University of Colombo, Sri Lanka
balachandran.24@cse.mrt.ac.lk, rtuthaya@cse.mrt.ac.lk
rpn@ucsc.cmb.ac.lk, arw@ucsc.cmb.ac.lk

Abstract

Emotion analysis plays a significant role in understanding human behavior and communication, yet research in Tamil language remains limited. This study focuses on building an emotion classifier for Tamil texts using machine learning (ML) and deep learning (DL), along with creating an emotion-annotated Tamil corpus for Ekman's basic emotions. Our dataset combines publicly available data with re-annotation and translations. Along with traditional ML models we investigated the use of Transfer Learning (TL) with state-of-the-art models, such as BERT and Electra based models. Experiments were conducted on unbalanced and balanced datasets using data augmentation techniques. The results indicate that Multinomial Naive Bayes (MNB) and Support Vector Machine (SVM) performed well with TF-IDF and BoW representations, while among Transfer Learning models, LaBSE achieved the highest accuracy (63% balanced, 69% unbalanced), followed by TamilBERT and IndicBERT.

1 Introduction

Emotional Analysis (EA), an extended version of sentiment analysis (SA), extracts emotions from human output using physiological qualities like voice, looks, hand motions, body developments, heartbeat and blood pressure (Chuang and Wu, 2004). R. W. Picard emphasized computers must understand emotions for effective human-computer interaction (Picard, 1997). In this digital era, the divide between ethnic groups and communities has diminished where people love to communicate, understand and experience diversity. Computer language translation plays a prominent role here though it can lead to misinterpretations of emotions within the context on certain occasions. Ze-Jing Chuang and Chung-Hsien Wu's Multi-Modal Emotion Recognition research combining speech and text produced better results than either input alone (Chuang and Wu, 2004). They created the textual model by defining keywords for emotions with emotion modification values. EA remains complex when processing only textual data compared to speech and vision.

To proceed with a structured analysis, selecting a reliable and widely accepted emotional categorization framework is essential. Ekman's basic emotions—Anger, Disgust, Fear, Joy, Sadness, and Surprise caters the above requirement in the research community (Ekman, 1992). Proposed by psychologist Paul Ekman, this scheme was developed based on cross-cultural studies that demonstrated these emotions as universal. Also the mentioned schema found to be consistently recognizable across different societies. The framework has proven highly valuable in emotion recognition tasks for human-computer interaction systems, social robotics, and content analysis.

Creating a well-annotated emotion dataset with Ekman's basic emotions spectrum poses a major challenge for Tamil language EA, as existing datasets exhibit significant class imbalances as well as the emotions not directly aligning with the schema. The TamilEmo dataset (Vasantharajan et al., 2021) requires reclassification with linguistic expert input for Ekman's basic emotions and potential re-annotation to validate the classification. While ACTSEA (Jenarthanan et al., 2019) is not fully publicly available with less than 500 samples, this research aligns with Ekman's basic emotions. The rest of the sentences that do not belong to any of the above is considered Neutral. With current state of the art (SOTA) transformer-based approaches, a proper dataset with class balance will significantly contribute to this research and result in better accuracy than previous works (Vasantharajan et al., 2021; Gokhale et al., 2022). This research aspires to create a balanced Tamil emotion annotated corps and improve emotion detection/recognition by using Natural Language Processing (NLP) with

Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 17–30

Machine Learning (ML) and Deep Learning (DL) techniques.

2 Related Works

Emotions can be understood through punctuation, catchphrases, syntax, and semantic data (Chuang and Wu, 2002). The SNoW learning architecture outperformed baseline Naive model and Bag Of Words (BoW) approach (Alm et al., 2005), while Wu et al. presented automatic emotion recognition through semantic labels and attributes (Wu et al., 2006).

A hybrid keyword-based and learning-based approach using SVM achieved 96.43% accuracy (Binali et al., 2010). Shivhare proposed an Ontology method based on commonsense knowledge and interrelationship between entities and core vocabulary (Shivhare and Khethawat, 2012). For Japanese earthquake-related tweets, Vo B and Collier N concluded that simple N-gram features performed best using MNB model (Vo and Collier, 2013).

Canales L and Martinez-Barco's survey discussed computational approaches categorized as lexicon-based and ML-based, noting keyword-based approaches (Strapparava and Mihalcea, 2008), ontology-based (Shivhare and Khethawat, 2012) and statistical approaches (Chuang and Wu, 2002) as lexical methods. Their findings showed keyword-based approaches yield higher accuracy, while supervised learning outperforms unsupervised methods despite requiring resource-intensive annotated datasets (Canales and Martínez-Barco, 2014). SVM has been a traditional supervised learning technique for EA (Hakak et al., 2017), though Nasir A et al. found MNB models perform better than SVM, decision tree algorithm and k-nearest neighbour methods (Ab. Nasir et al., 2020).

2.1 Emerging of Transformers

The introduction of transformers revolutionized the DL field (Vaswani et al., 2017), with BERT becoming SOTA in many NLP implementations despite higher resource consumption (Devlin et al., 2018). Various BERT variants emerged, including mBERT and ALBERT, while XLM-RoBERTa later outperformed mBERT (Conneau et al., 2019b).

Electra emerged as a resource-efficient alternative to BERT (Clark et al., 2020), while Huang C et al.'s ensemble method combining HRLCE and BERT achieved a macro-F1 score of 0.7709 (Huang et al., 2019). Yang K et al. enhanced pre-trained models using MLM and NSP (Yang et al., 2019). Al Omari H, Abdullah M and Shaikh S executed a dual model using BiLSTM and BERT, resulting in an F1 score of 0.748 (Al-Omari et al., 2020). Acheampong F et al.'s review recommends exploring more BERT variants and ensemble models (Acheampong et al., 2021).

Comparative analyses show BERT and Electra outperform RoBERTa, XLM-R and XLNet in fine-grained emotions detection with lower training time (Frye and Wilson, 2022), though Cortiz D found DistillBERT, RoBERTa and XLNet superior to Electra (Cortiz, 2021). Zhang S, Yu H and Zhu G's Electra-based model with attention mechanism and BiLSTM achieved mean accuracies of 94.657 and 93.713 for Chinese language emotion detection (Zhang et al., 2022).

2.2 Research on Tamil Language

Tamil language EA research remains limited compared to other languages. For sentiment analysis, Padmamala R and Prema V's RNN approach achieved 71.1% accuracy (Padmamala and Prema, 2017), while Shanmugavadivel K et al.'s CNN with Bi-LSTM achieved 0.66 accuracy on tamil code-mixed texts (Shanmugavadivel et al., 2022). Sajeetha T's experiments with multiple approaches achieved 79% accuracy using fastText (Thavareesan and Mahesan, 2019), later improving to 88% accuracy using Word2vec and fastText with rule-based approach (Thavareesan and Mahesan, 2020). Sharmista's product review sentiment analysis concluded that ensemble methods produced optimal results (Ramaswami, 2020).

2.3 Research on Tamil Emotion Analysis

Dakshina k and Sridhar R's LDA-based emotion recognition for Tamil songs achieved 72% accuracy using supervised learning with 160 songs and 5 annotators (Dakshina and Sridhar, 2014). Charangan V et al.'s TamilEmo corpus classified 31 emotions from 42,686 sentences scraped from YouTube comments. These samples were annotated with an inter-annotator agreement of 0.7452. A major concern in the dataset is the class imbalance among emotion categories, with the emotion "admiration" having the highest sample count of 6,682 samples, while the emotion "desire" has the lowest sample count, with only 208 samples. Their ML methods achieved a maximum 0.42 F1 score (Vasantharajan et al., 2021). Gokhale O et al. attempted transformer ensemble method and could not achieve

significant improvements for the very same dataset (Gokhale et al., 2022).

Rajalakshmi et al.'s investigation of emoji impact in Tamil Texts showed that replacing emojis with keywords performed best, followed by emoji-present and emoji-removed approaches. Their TF-IDF and XGBoost combination outperformed the MuRIL pre-trained model (Rajalakshmi et al., 2022). This shows that containing the emojis in the dataset is essential for higher results.

3 Dataset Overview

In this study, we adopted Ekman's basic emotions: anger, disgust, fear, joy, sadness, surprise, and neutral since they are widely accepted and frequently used in emotion research, especially in high-resource languages. This approach is a standard in emotion classification tasks, making it a suitable framework to extend to the Tamil language, where similar work has been limited. By aligning with this framework, we aim to standardize emotion classification in Tamil and provide a valuable resource for future research.

3.1 Data Collection

TamilEmo(Vasantharajan et al., 2021) was the only publicly available dataset which was emotion annotated in Tamil language. The TamilEmo dataset had 31 emotion classes and they were grouped into seven primary emotions: Hope, Neutral, Love, Bewilderment, Disgrace, Pathos and Laughter. Of these seven emotions, only three could be mapped directly to Ekman's basic emotions as $Neutral \longrightarrow Neutral$, $Pathos \longrightarrow Sad$ and $Laughter \longrightarrow Joy$. For the other emotions, we had to go with the fine-grained emotions of 31 classes.

Emotions	Angry	Disgust	Fear	Joy	Neutral	Sad	Surprise	Unclassified
admiration	☐	☐	☐	☑	☑	☐	☑	☐
amusement	☐	☐	☐	☑	☑	☐	☐	☐
anger	☑	☐	☐	☐	☐	☐	☐	☐
annoyance	☑	☑	☐	☐	☑	☐	☐	☐
anticipation	☐	☐	☐	☐	☐	☐	☐	☑
approval	☐	☐	☐	☐	☐	☐	☐	☑
caring	☐	☐	☐	☐	☐	☐	☐	☑
confusion	☐	☐	☐	☐	☐	☐	☐	☑
curiosity	☐	☐	☐	☑	☑	☐	☑	☐
desire	☐	☐	☐	☐	☐	☐	☐	☑
disappointment	☑	☐	☐	☐	☑	☑	☐	☐
disapproval	☐	☐	☐	☐	☐	☐	☐	☑
disgust	☐	☑	☐	☐	☐	☐	☐	☐
embarrassment	☐	☑	☐	☐	☑	☑	☐	☐
excitement	☐	☐	☐	☑	☑	☐	☐	☐
fear	☐	☐	☑	☐	☐	☐	☐	☐
gratitude	☐	☐	☐	☑	☑	☐	☑	☐
grief	☐	☐	☐	☐	☐	☑	☐	☐
joy	☐	☐	☐	☑	☐	☐	☐	☐
love	☐	☐	☐	☑	☑	☑	☐	☐
nervousness	☐	☐	☑	☐	☑	☐	☐	☐
neutral	☐	☐	☐	☐	☑	☐	☐	☐
optimism	☐	☐	☐	☑	☑	☐	☑	☐
pride	☐	☐	☐	☑	☑	☐	☑	☐
realization	☐	☐	☐	☐	☐	☐	☐	☑
relief	☐	☐	☐	☑	☐	☐	☐	☐
remorse	☐	☐	☐	☐	☐	☑	☐	☐
sadness	☐	☐	☐	☐	☐	☑	☐	☐
surprise	☐	☐	☐	☐	☐	☐	☑	☐
teasing	☐	☐	☐	☑	☐	☐	☐	☐
trust	☐	☐	☐	☐	☐	☐	☐	☑

Figure 1: Emotion Mapping of TamilEmo Dataset

Under the guidance of our language expert panel, these 31 classes of emotions were mapped to the emotion categories. If any of the emotions were ambiguous and could not be directly mapped, so they were classified into mixed emotions. Few of them couldn't be concluded to any set of emotions or could fall under more than three emotions, so they were categorized as unclassified. All the mixed emotions were included under neutral emotion as well to be sure when annotating. Figure 1 shows how we categorized the emotions. When validating the samples with corresponding emotions, we learned that many samples had been contradictorily annotated in the original study. Another main issue with this dataset is the class imbalance. The emotion admiration has 6682 samples, and the emotion desire has only 208 samples. It was understood that the samples in this dataset would not be sufficient to have a balanced dataset. Therefore we had options to scrape data from the web with keywords and annotate or translate an available English dataset to Tamil and validate them.

We used an English Emotion Dataset (Saravia et al., 2018), which is publicly available in Kaggle and Huggingface as our secondary dataset. This dataset contained English Twitter messages with six basic emotions: anger, fear, joy, love, sadness, and surprise. Except for the emotion of love, all the other emotions directly corresponded to our study. The emotions distributions can be seen in Figure 2

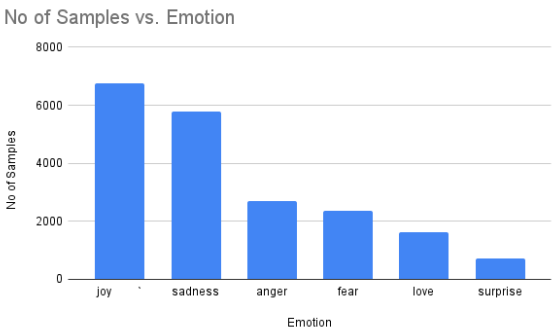

Figure 2: Class Distribution - English Emotion Dataset

3.2 Data Annotation

The datasets were divided into chunks of a maximum of 1000 - 1100 samples to make it easier for the annotators. Every sample was annotated by a pair of annotators using separate spreadsheets without any influence from each other. The annotators were instructed to disregard sarcasm and interpret sentences by their literal meaning since

satire is considered out of scope in our study. The native Tamil-speaking undergraduates of the University of Colombo School of Computing were the annotators.

The samples which were explicitly categorized as in the original study were annotated by selecting whether it is correctly classified or misclassified. For the other emotions, the possible emotions were listed in a drop-down and the annotators were asked to choose the best option. When both annotators completed their annotations, the results were compared, and a third one annotated the contradicting samples. Then the maximum of the emotions selected was made final. In some instances, all three annotators' choices differed from each other. In that case, those samples were filtered out from the final dataset.

The English Emotion dataset was combined as one single dataset CSV file which was initially divided into train, test and validation datasets. Translation was done using Google translate and the annotators were asked to annotate whether the translation made sense or not by selecting either yes or no. Then they were also asked to give points according to the samples giving justice to the emotions. When the pair of annotators had done their work, and checked whether both agreed that the translation was correct and whether the point total was above half of the maximum value. If they contradicted the translation, a third annotator validated those samples. If most of them selected "yes" for the translation, then again as before, the total points were checked for more than half the maximum value. Then the samples were finalized to their corresponding emotions and the rest were abandoned. The point format is as follows.

- 0: Does not align with the emotion.
- 1-4: Have some context related to the emotion, but the translation of the sentence is not appropriate (the overall sentence does not make sense).
- 5: Have context related to the emotion, but the translation is ambiguous, which might exhibit mixed emotions.
- 6-10: Have descent alignment with the emotion.

As this whole annotation process is manual it was a huge concern. The time taken to annotate was longer than anticipated, and it was not easy to manage the annotators. These are the few main challenges we faced during this phase.

- Due to pair wise annotation, the datasets could be annotated at a rate of half the annotators only.
- Inconsistency and slow process of few annotators, where continuous annotation of the next dataset assigned to them was not possible with everyone.
- Have to wait for both the annotators to finish annotating so we can validate it with the third annotator.
- For certain samples, the annotators were not able to conclude their results to a specific emotion. Those samples were finally excluded.

After continuous annotations and validations, at the end approximately more than 50,000 annotations were completed and the final dataset ended with 16804 samples. Figure 3 depicts the final dataset overview.

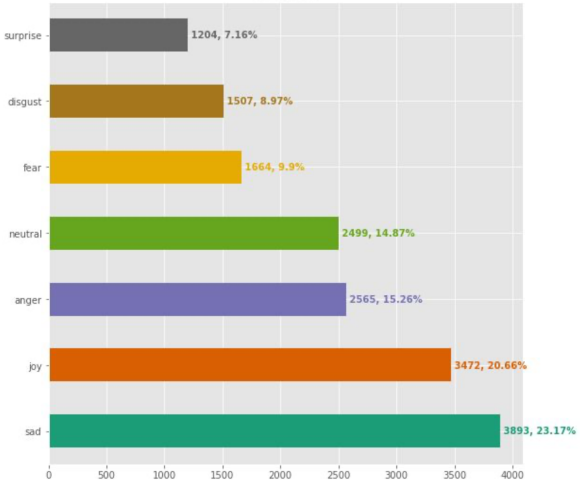

Figure 3: Final Dataset Overview

A balanced dataset could not be generated in the last stage as planned. Several factors contributed to this such as, inherent class imbalance, language-specific challenge, annotation challenges, inherent imbalance in real-world data and time constraints.

The following are the Average Cohen's Kappa values for the annotations of all datasets in Table 1.The average inter-annotator agreement between annotators A and B: 72%, B and C: 69%, A and C: 79% and the average of all inter-annotator agreements is 73%.

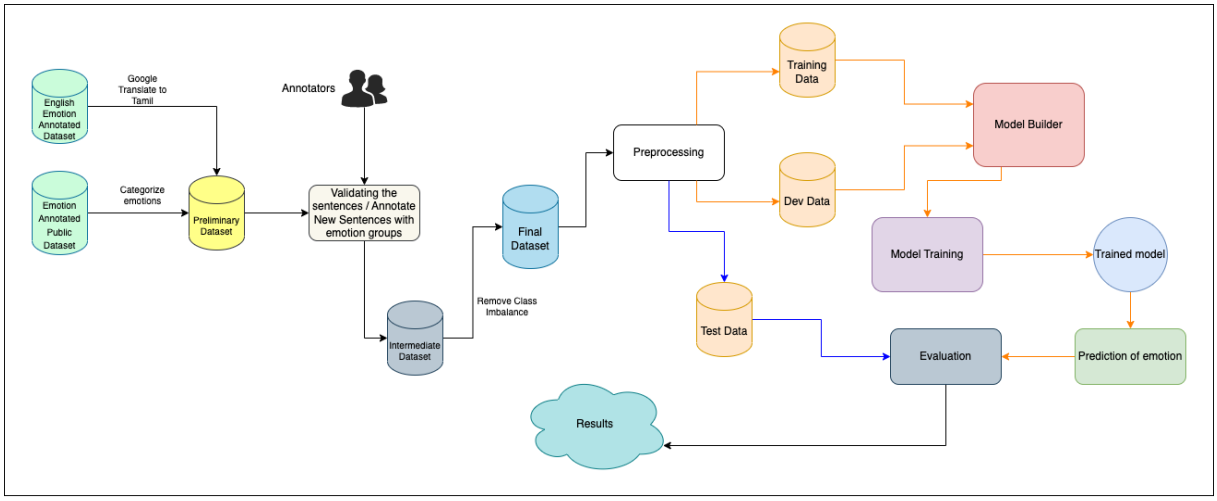

Figure 4: High level approach of the study

	A ↔ B	B ↔ C	A ↔ C	Average
Average Kappa for Annotators	72%	69%	79%	73%

Table 1: Kappa statistics Average for the Final Dataset

4 Experiments and Evaluation

Figure 4 describes the methodology of the research in a higher level. The following sections have detailed information on the experiments and the results.

4.1 Preprocessing

Several measures were taken during the preprocessing phase to ensure the quality and consistency of our dataset. First, the hyperlinks were deleted and profile tags, as well as the white spaces, because they do not contribute to the emotional substance of the text. We also chose against eliminating punctuation and emoticons because they can considerably alter the emotions represented in the samples. Similarly, English terms were not eliminated because they, too, could reflect feelings in some cases.

4.2 Models Utilized

Both ML and DL approaches were employed in the study. In addition to that dual model approach of involving pre-trained models and deep learning models were also experimented. In the traditional ML experiments SVM, random forest, naive bayes, multinomial naive bayes (MNB), decision tree, passive aggressive and K-Nearest Neighbour (KNN) were included. In the other hand for DL models, Universal Sentence Encoder CMLM - Multilingual Base (Yang et al., 2021), MuRIL-Large (Multilingual Representations for Indian Languages) (Khanuja et al., 2021), BERT Multilingual Cased (bert_multi_cased_L-12_H-768_A-11) (Devlin et al., 2018), XLM-R0BERTa Multilingual Cased (xlm_roberta_multi_cased_L-24_H-1024_A-16) (Conneau et al., 2019a), DistilBERT (distilbert_multi_cased_L-6_H-768_A-12) (Sanh et al., 2019), Tamillion (An Electra based monolingual Tamil pre-trained model), IndicBERT, LaBSE (Language-agnostic BERT Sentence Encoder) (Kakwani et al., 2020), LaBSE (Language-agnostic BERT Sentence Encoder) (Feng et al., 2020), TamilBERT(Joshi, 2022) were utilized. Then as mentioned for dual model approach, CNN with Bi-LSTM, CNN with Bi-GRU and LaBSE with Bi-LSTM were combined. The architecture and training source details of the models are attached in the Appendix A

4.3 Unbalanced Final Dataset

When rendering the results, the best-performing models in this experiment appeared to be SVM and Passive Aggressive, especially when combined with TF-IDF Unigram with Bigram and FastText text representations. The performance of Naive Bayes and Multinomial Naive Bayes models varied significantly depending on the text representation used. They performed better with Bag of Words Unigram and Bigram representations, consistent with the previous experiments. However, their performance dropped when using other text representations like the Bag of Words Trigram and TF-IDF Unigram. This is illustrated in Table 2, where the best-performing text representations for each model are highlighted in yellow, and the highest-performing models for each text representation is

indicated in bold.

Decision Tree, K Nearest Neighbor, and Random Forest models showed modest improvements in accuracy in the unbalanced dataset experiment compared to their performance in the previous downsampled dataset experiment. These models might have been less sensitive to class imbalance, and their performance could have depended more on the quality and quantity of the available data.

	Naive Bayes	MNB	SVM	Decision Tree	KNN	Random Forest	Passive Aggressive
Bag of Words							
Unigram	0.16	**0.53**	**0.52**	0.44	0.39	**0.51**	0.44
Bigram	0.36	**0.49**	**0.50**	0.45	0.31	**0.48**	0.47
Trigram	0.34	0.39	**0.43**	0.36	-	0.37	0.36
TF-IDF							
Unigram	0.30	0.34	**0.44**	0.34	0.20	0.37	**0.44**
Unigram with Bigram	0.36	0.47	**0.58**	0.45	0.24	0.53	0.54
FastText	0.40	-	**0.56**	0.37	0.44	0.53	**0.56**

Table 2: Accuracy comparison of ML Models for the Final Unbalanced Dataset

In terms of accuracy, most models showed an increase in performance when using the unbalanced dataset compared to the downsampled one. This might have been due to the larger amount of data available for training, which generally helps the models better understand the patterns and capture more nuanced relationships between the features and target emotions. Moreover, the class imbalance in the unbalanced dataset might have also played a role in the increased accuracies since the models were now better exposed to the majority class, which is more frequently seen in real-life scenarios.

Model	Accuracy
Universal Sentence Encoder	61%
MuRIL-Large	60%
BERT Multilingual Cased	51%
XLM-R0BERTa Multilingual Cased	43%
Distil-BERT	49%
Tamillion	62%
IndicBERT	64%
TamilBERT	67%
LaBSE	**69%**

Table 3: Accuracy comparison of Transformer Models for the Final Unbalanced Dataset

When comparing models focused on one language (monolingual) and models that worked with multiple languages (multilingual), we found that TamilBERT and Tamillion (monolingual models) had higher accuracies. This could have been because they were designed specifically for Tamil. However, LaBSE, a multilingual model, also performed very well, achieving an accuracy of 69%.

Among Indian language-based models, IndicBERT, which was trained on 12 major Indian languages, achieved an accuracy of 64%. This indicates that a model trained on multiple Indian languages can still perform well for Tamil language classification. When examining different model types, BERT-based models like TamilBERT, LaBSE, and BERT Multilingual Cased showed varying levels of success. TamilBERT and LaBSE performed significantly better. The ELECTRA-based model, Tamillion, also produced good results with an accuracy of 62%. As mentioned in the literature, ELECTRA models tend to perform well, but we were unable to find more pre-trained ELECTRA models related to Tamil for experimentation. Finally, the ALBERT-based model, IndicBERT, demonstrated strong performance with an accuracy of 64%.

In the current experiment, we observed a significant improvement in the accuracies of the transfer learning models compared to the machine learning models on the unbalanced dataset. LaBSE and TamilBERT achieved higher accuracies of 69% and 67% respectively, which were considererably higher than the best machine learning model using TF-IDF Unigram with Bigram (58%). This highlights the advantages of utilizing pre-trained models.

Model	Accuracy
CNN with Bi-LSTM	56%
CNN with Bi-GRU	56%
LaBSE with Bi-LSTM	**61%**

Table 4: Accuracy comparison of Dual Models for the Final Unbalanced Dataset

Upon examining the outcomes of the hybrid model approach applied to the same dataset, it became evident that their effectiveness lay somewhere between the machine learning and transfer learning models. The CNN combined with a Bi-LSTM model reached an accuracy of 56%, while the CNN paired with a Bi-GRU model achieved a 56% accuracy rate. The LaBSE, alongside a Bi-LSTM model incorporating transfer learning, produced a higher accuracy level of 61%. Despite the enhancements displayed by the hybrid models compared to the machine learning models, they failed to outperform transfer learning models like the fine-tuned models.

4.4 Balanced Final Dataset

To achieve a balanced dataset, a decision was made to split the original samples into training and test sets before proceeding with data augmentation. The split involved allocating 15% of the number of samples in the minority class (241 samples) as the test data, while the remaining samples were retained for training. It was determined that 2250 samples per emotion would be used for the training data.

Referring to the work of Jie and Gao (Gao, 2020) on Data Augmentation in Solving Data Imbalance Problems, it was found that translation proved to be an effective technique for upsampling textual data. Based on this finding, translation was chosen as the upsampling technique. Google Translate was utilized for the translation process. Initially, the dataset was translated from Tamil to English and then back to Tamil. Following the translation, a careful selection process was implemented to ensure that the majority of the samples remained original, with only the necessary number of samples required for balancing the dataset being included from the translated samples. These samples were randomly selected from each class within the translated set.

	MNB	SVM	Random Forest	Passive Aggressive
Bag of Words				
Unigram	**0.47**	**0.46**	0.45	0.39
Bigram	**0.45**	0.43	0.43	0.41
TF-IDF				
Unigram	**0.40**	**0.41**	0.34	**0.40**
Unigram with Bigram	**0.48**	**0.48**	**0.48**	0.44
FastText	-	**0.47**	0.44	0.44

Table 5: Accuracy comparison of ML Models for the Final Balanced Dataset

The results of this experiment indicate that the models performed below expectations compared to the machine learning experiments conducted on the unbalanced dataset, with the exception of Multinomial Naive Bayes combined with TF-IDF Unigram with Bigram text representation, which showed a slight improvement. When compared to the previous balanced dataset experiment, the differences in results were relatively minor. The models performed better overall compared to the earlier experiments, except for SVM, Random Forest, and Passive Aggressive with the text representations TF-IDF Unigram with Bigram and FastText. However, when compared to the deep learning (DL) models, the performance of these models fell significantly below the average.

In this experiment, when analyzing the results exclusively, TF-IDF Unigram with Bigram emerged as the top-performing text representation, achieving accuracy ranging from 44% to 48%. Similarly, Multinomial Naive Bayes stood out among the models, also with accuracy ranging from 44% to 48%. The Bag of Words (BoW) Unigram and SVM model provided tough competition to the top-ranking models.

Model	**Accuracy**
CNN with Bi-LSTM	52%
CNN with Bi-GRU	50%
LaBSE with Bi-LSTM	55%
Tamillion	57%
TamilBERT	62%
LaBSE	**63%**

Table 6: Accuracy comparison of Transformer and Dual Models for the Final Balanced Dataset

As expected, the deep learning (DL) models performed better than the machine learning (ML) models. Among the hybrid models, LaBSE with Bi-LSTM achieved an accuracy of 55%, which was higher than the other models. CNN with Bi-LSTM outperformed CNN with Bi-GRU, while in the unbalanced dataset, they performed at similar levels. The transfer learning model, Tamillion, achieved an accuracy of 57%, which was relatively lower than the bert-based models LaBSE and TamilBERT. Among the fine-tuned combinations, LaBSE achieved the highest accuracy of 63%. Appendix B describes this best performing model architecture. It is worth noting that these transfer learning models outperformed some of the other transfer learning models from the unbalanced dataset, even though the accuracy of all the models for this balanced dataset dropped compared to the previous experiment.

4.5 Error Analysis

For each model outputs, we did observation study on the confusion matrix followed by the error analysis. The observation pointed out that the confusion matrices produced throughout the experiment were quite similar, which suggests that the dataset maintains internal consistency. The Confusion matrix of the best model LaBSE is displayed in Figure 5. When observing the confusion matrix, commonly Disgust is confused with Anger and Joy with Neutral even with the balanced dataset. 105 samples

of Disgust have been falsely predicted as Anger and 57 vice versa. 48 samples of Joy have been incorrectly predicted as Neutral, and 41 samples in the other way. Except for Fear and Sadness, other emotions have some confusion with Neutral emotion, which is obvious that these emotions might also tend to be neutral on certain occasions.

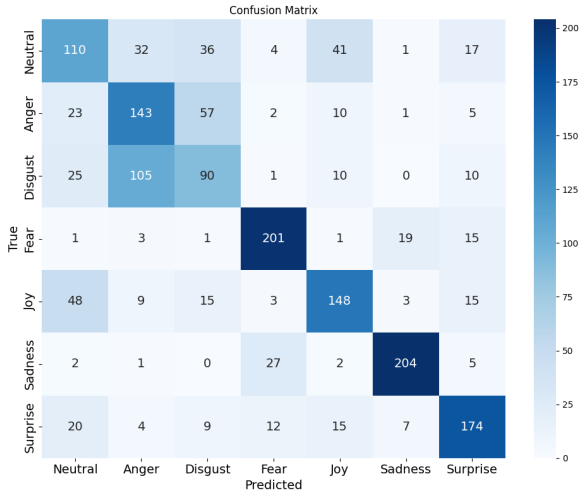

Figure 5: Confusion Matrix of LaBSE model for the Final Balanced Dataset

The most common thing observed throughout this experiment is that the emotion Disgust is being confused with Anger and Neutral with Joy. Since the emotion Disgust is the minority class with a low number of 241 samples, and the confused number of samples is significant and has a huge impact. So an error analysis was done using the LIME library for the emotion Disgust, which was predicted as Anger, and the analysis is attached in Appendix C. As per the analysis result, the model predicted this as anger with 97% confidence and showed Disgust as 0.02% only. So when we look at this example, the sentence can also actually be said as it represents Anger because of the presence of words expressing 'killing' and 'cursing'. This particular sentence has some words relating to castism and extremism based on caste, which might have led the annotators to annotate it as Disgust which also makes sense. Because the samples in Disgust or Anger get confused with each other, they may be exhibiting both emotions in a way, these confusions may have occurred, and the results are affected correspondingly.

5 Discussion

Throughout the study, various text representation techniques have been utilized. TF-IDF Unigram with Bigram yielded higher results in almost all our experiments. Other than these FastText pre-trained models also gave fairly competitive results to the above representations. Considering the transfer learning models, their own preprocessor should be used to achieve better results.

Out of the models that were trained, SVM scored the best and consistently gave better results throughout the ML experiments, especially combined with TF-IDF Unigrams with Bigrams. MNB, Random Forest and Passive Aggressive can be considered alternative models with slightly below-par performance.

Under transfer learning models, the LaBSE model is the highest achieving model with an accuracy 69% for the unbalanced and 63% for the balanced dataset after fine-tuning. Along with that, TamilBERT and IndicBERT also gave fair results. The Electra-based Tamillion model did not perform as expected from the literature (Zhang et al., 2022), where Electra models gave a nearly par performance value with the BERT models. Finally, considering the hybrid models, LaBSE combined with Bi-LSTM models has served better in the hybrid category. Out of all the model categories, transfer learning approaches outperformed every other category.

We compared our results with the original study, where our results can be compared with their **7-class group results**. So far in this experiment, **TF-IDF with Unigrams and Bigrams** combining the **MNB model** for our final balanced dataset of **15,750 samples** has performed **F1-Score of 0.46**, which is higher than the original study. The size of the dataset also matters when comparing the results, as well as the class distribution. The smaller dataset might also be a reason for us reaching higher results. Since the original study employs an **unbalanced dataset**, it is fair to compare the results of our unbalanced dataset. Our best-performing model, **LaBSE** scored an **F1-Score of 0.64** while **TamilBERT**, **IndicBERT**, and **Tamillion** scored **0.62**, **0.60**, and **0.57**, respectively, which are **significantly better results than the original study**.

6 Limiations and Future work

The limitations include the scarcity of quality emotion-annotated datasets for the Tamil language, especially for Ekman's basic emotions. Along with that, the fine-grained nature of the available dataset which did not align with Ekman's basic emotions

also led to complete re-annotation. Additionally, due to class imbalance, the emotion with the highest sample count had to be reduced to 2,250, enabling decent augmentation for the lowest sample count, which resulted in an overall reduction in total sample count.

As a continuation of this study, there is a vast scope to experiment with rule-based preprocessing where negation words and word polarity can be considered. This can be extended to code mixed corpus as well. Investigating more combinations of hybrid models and ensemble approaches with the trained models might give better results. On top of this contrastive learning as well employment of large language models leveraging prompt engineering can be considered. Identifying sarcasm is another dimension to explore in this domain. This textual classification can be integrated with speech-to-text jobs and evolve into an emotion classifier for speech. Other than Ekman's basic emotions, the writing styles such as formal, casual etc. can also be classified.

Acknowledgment

A heartfelt thanks to the authors of "TamilEmo: Finegrained Emotion Detection Dataset for Tamil for providing us with their dataset to conduct my research. Our sincere gratitude for my advisor and language expert Mr V. Vimalathithan for his invaluable guidance and feedback. Special thanks go to the native Tamil students of University of Colombo School of Computing who took part in annotating the dataset.

References

Ahmad Fakhri Ab. Nasir, Eng Nee, Chun Sern Choong, Ahmad shahrizan Abdul ghani, Anwar P P Abdul Majeed, Asrul Adam, and Mhd Furqan. 2020. Text-based emotion prediction system using machine learning approach. *IOP Conference Series: Materials Science and Engineering*, 769:012022.

Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. 2021. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, 54:5789–5829.

Hani Al-Omari, Malak A. Abdullah, and Samira Shaikh. 2020. Emodet2: Emotion detection in english textual dialogue using bert and bilstm models. pages 226–232. Institute of Electrical and Electronics Engineers Inc.

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: machine learning for text-based emotion prediction.

Haji Binali, Chen Wu, and Vidyasagar Potdar. 2010. Computational approaches for emotion detection in text.

Lea Canales and Patricio Martínez-Barco. 2014. Emotion detection from text: A survey.

Ze-Jing Chuang and Chung-Hsien Wu. 2002. Emotion recognition from textual input using an emotional semantic network.

Ze-Jing Chuang and Chung-Hsien Wu. 2004. Multi-modal emotion recognition from speech and text.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Unsupervised cross-lingual representation learning at scale.

Diogo Cortiz. 2021. Exploring transformers in emotion recognition: a comparison of bert,distillbert, roberta,xlnetandelectra.

K. Dakshina and Rajeswari Sridhar. 2014. Lda based emotion recognition from lyrics. volume 27, pages 187–194. Springer Science and Business Media Deutschland GmbH.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova Google, and A I Language. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Paul Ekman. 1992. Are there basic emotions?

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *Preprint*, arXiv:2007.01852.

Robert H Frye and David C Wilson. 2022. Comparative analysis of transformers to support fine-grained emotion detection in short-text data.

JIE Gao. 2020. Data augmentation in solving data imbalance problems.

Omkar Gokhale, Shantanu Patankar, Onkar Litake, Aditya Mandke, and Dipali Kadam. 2022. Optimize$_prime$@$dravidianlangtech - acl2022 : Emotionanalysisintamil$.

Nida Manzoor Hakak, Mohsin Mohd, Mahira Kirmani, and Mudasir Mohd. 2017. Emotion analysis: A survey. In *2017 international conference on computer, communications and electronics (COMPTELIX)*, pages 397–402. IEEE.

Chenyang Huang, Amine Trabelsi, and Osmar R. Zaïane. 2019. Ana at semeval-2019 task 3: Contextual emotion detection in conversations through hierarchical lstms and bert.

Rajenthiran Jenarthanan, Yasas Senarath, and Uthayasanker Thayasivam. 2019. *ACTSEA: Annotated Corpus for Tamil Sinhala Emotion Analysis*.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages. *Preprint*, arXiv:2103.10730.

R. Padmamala and V. Prema. 2017. Sentiment analysis of online tamil contents using recursive neural network models approach for tamil language. In *2017 IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, pages 28–31.

R W Picard. 1997. Affective computing.

Ratnavel Rajalakshmi, Faerie Mattins R, Srivarshan Selvaraj, Antonette Shibani, Anand Kumar M, and Bharathi Raja Chakravarthi. 2022. Understanding the role of emojis for emotion detection in Tamil. In *Proceedings of the First Workshop on Multimodal Machine Learning in Low-resource Languages*, pages 9–17, IIIT Delhi, New Delhi, India. Association for Computational Linguistics.

M Ramaswami. 2020. Sentiment analysis on tamil reviews as products in social media using machine learning techniques: A novel study doctor of philosophy in computer science.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

Kogilavani Shanmugavadivel, Sai Haritha Sampath, Pramod Nandhakumar, Prasath Mahalingam, Malliga Subramanian, Prasanna Kumar Kumaresan, and Ruba Priyadharshini. 2022. An analysis of machine learning models for sentiment analysis of tamil code-mixed data. *Computer Speech Language*, 76:101407.

Shiv Naresh Shivhare and Prof Saritha Khethawat. 2012. Emotion detection from text.

Carlo Strapparava and Rada Mihalcea. 2008. Learning to identify emotions in text. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 1556–1560.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2019. *Sentiment Analysis in Tamil Texts: A Study on Machine Learning Techniques and Feature Representation*.

Sajeetha Thavareesan and Sinnathamby Mahesan. 2020. *Sentiment Lexicon Expansion using Word2vec and fastText for Sentiment Prediction in Tamil texts*.

Charangan Vasantharajan, Sean Benhur, Prasanna Kumar Kumarasen, Rahul Ponnusamy, Sathiyaraj Thangasamy, Ruba Priyadharshini, Thenmozhi Durairaj, Kanchana Sivanraju, Anbukkarasi Sampath, Bharathi Raja Chakravarthi, and John Phillip McCrae. 2021. Tamilemo: Finegrained emotion detection dataset for tamil.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Bao-Khanh H Vo and Nigel Collier. 2013. Twitter emotion analysis in earthquake situations.

Chung-Hsien Wu, Ze-Jing Chuang, and Yu-Chung Lin. 2006. Emotion recognition from text using semantic labels and separable mixture models.

Kisu Yang, Dongyub Lee, Taesun Whang, Seolhwa Lee, and Heuiseok Lim. 2019. Emotionx-ku: Bert-max based contextual emotion classifier.

Ziyi Yang, Yinfei Yang, Daniel Cer, Jax Law, and Eric Darve. 2021. Universal sentence representation learning with conditional masked language model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6216–6228, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shunxiang Zhang, Hongbin Yu, and Guangli Zhu. 2022. An emotional classification method of chinese short comment text based on electra. *Connection Science*, 34:254–273.

A Overview of the Transfer Learning Models Used

Table 7: Overview of the Transfer Learning Models

Model	Language Type	Architecture	Preprocessor
Universal Sentence Encoder CMLM - Multilingual Base (Yang et al., 2021)	Universal sentence encoder for 100+ languages trained with a conditional masked language model.	The base model employs a 12-layer BERT transformer architecture.	universal-sentenc e-encoder-cmlm/ multi lingual-preprocess
MuRIL-Large (Multilingual Representations for Indian Languages) (Khanuja et al., 2021)	Pre-trained on 17 Indian languages, and their transliterated counterparts.	A BERT Large (24L) model	MuRIL_pre process
BERT Multilingual Cased(Devlin et al., 2018)	Multilingual	BERT architecture. Uses L=12 hidden layers, a hidden size of H=768, and A=12 attention heads	bert_multi_case d_preprocess
XLM-R0BERTa Multilingual Cased(Conneau et al., 2019a)	Multilingual	Uses L=24 hidden layers, a hidden size of H=1024, and A=16 attention heads	xlm_roberta_mu lti_cased_pre process
Distil-BERT (Sanh et al., 2019)	Multilingual	Uses L=6 hidden layers, a hidden size of H=768, and A=12 attention heads	distilbert_multi_ cased_pre process
Tamillion	Monolingual (Tamil)	Model trained with Google Research's ELECTRA	ElectraTokenizer from transformers library
IndicBERT (Kakwani et al., 2020)	Multilingual. Pretrained exclusively on 12 major Indian languages	ALBERT (A Lite BERT for Self-supervised Learning of Language Representations) based model	AlbertTokenizer from transformers library
LaBSE (Language-agnostic BERT Sentence Encoder) (Feng et al., 2020)	Trained for sentence embedding for 109 languages	Based on the BERT architecture and uses a Siamese network with shared weights to learn a joint embedding space for different languages	BertTokenizer from transformers library **or** universal-sentenc e-encoder-cmlm/ multi lingual-preprocess
TamilBERT (Joshi, 2022)	Monolingual (Tamil)	Based on the BERT architecture	BertTokenizer from transformers library

B Best Model Architecture Details

The best performing model architecture consists of the Language-agnostic BERT Sentence Embedding (LaBSE) as the base model with a custom classification head. The complete architecture and training configuration are detailed below.

B.1 Model Architecture

- Input Layer: Text input layer accepting string data
- Base Model:
  - LaBSE Preprocessor
  - LaBSE Encoder
- Classification Head:
  - Dropout (rate = 0.2)
  - Dense Layer (128 units, ReLU activation)
  - Dropout (rate = 0.3)
  - Dense Layer (64 units, ReLU activation)
  - Dropout (rate = 0.1)
  - Output Layer (7 units, Softmax activation)

B.2 Training Configuration

- Optimizer: Adam
- Loss Function: Categorical Cross Entropy
- Early Stopping:
  - Monitor: Validation Loss
  - Training Duration: Stopped at epoch 5

B.3 Model Parameters

- Total Trainable Parameters: 109M*
- Base Model:
  - LaBSE Parameters: 109M*
- Classification Head:
  - Dense Layer 1: $128 \times \text{hidden\_size} + 128$ parameters
  - Dense Layer 2: $64 \times 128 + 64$ parameters
  - Output Layer: $7 \times 64 + 7$ parameters

C Error Analysis Using LIME

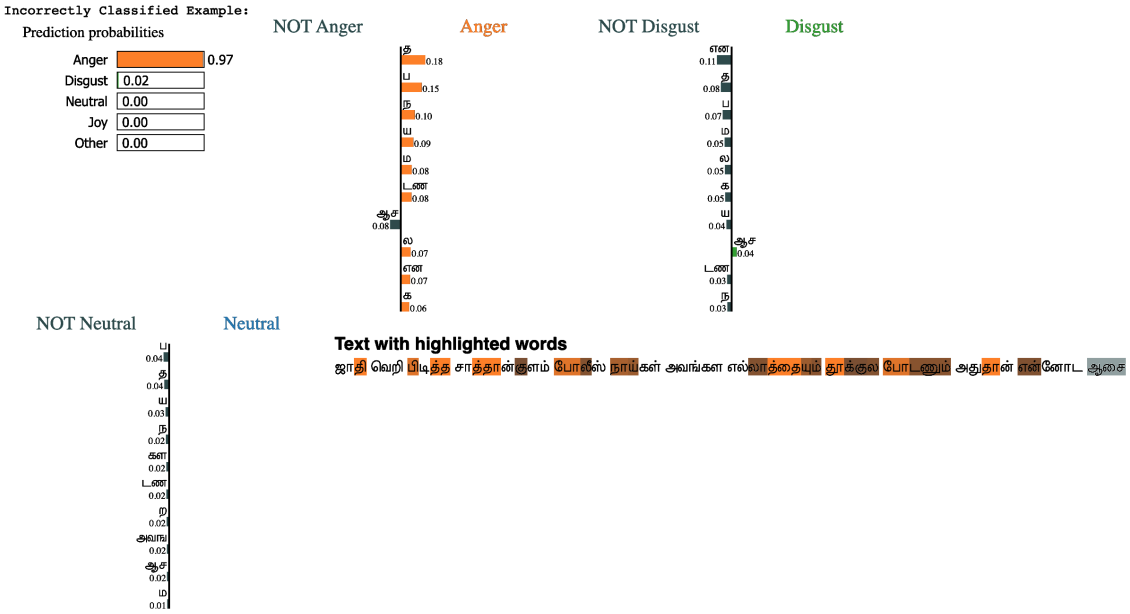

Figure 6: LIME Error Analysis - Disgust Predicted as Anger