Multimodal Sentiment Analysis of Tamil and Malayalam
D3

Abhinav Patil, Sam Briggs, Tara Wueger, D. D. O'Connell
Department of Linguistics, University of Washington
Seattle, WA
{abhinavp, briggs3, taraw28, danieloc}@uw.edu

Abstract

Sentiment analysis consists of categorizing a unit of language about a certain topic according to the author's attitude towards that topic. Our task is the classification of multi-modal data – text, audio, and video data – disseminated in the Dravidian languages Tamil and Malayalam into 5 separate sentiment classes: highly negative, negative, neutral, positive, and highly positive. We implement and evaluate multiple baseline models: a Multinomial Naive Bayes baseline model, a Logistic Regression model, and a Stochastic Gradient Descent Classifier Model for comparison. We also implement and evaluate a finetuned XLM RoBERTA model. To account for class imbalance, we use both naive resampling and SMOTE. We found that without resampling, both the baseline and the Neural Network have the same performance as a naive Majority Class Classifier baseline. However, with resampling, logistic regression and random forest both demonstrate gains over the baseline. Finally, we perform error analysis of the models.

1 Introduction

Sentiment analysis consists of categorizing a unit of language about a certain topic according to the author's attitude towards that topic. This task can be constructed in many ways, e.g. as a binary classification task (i.e., categorizing language in two classes, Positive and Negative sentiment), as an ordinal regression problem, or, as in our case, as a multi-class classification problem. Sentiment analysis has a rich history and many methods have been used (Cui et al., 2023).

2 Task Description

We will participate in the "Multimodal Abusive Language Detection and Sentiment Analysis: DravidianLangTech@RANLP 2023" shared task hosted on CodaLab (B. et al., 2023). [1]

This shared task has two subtasks: abusive language detection in Tamil, and sentiment analysis in both Tamil and Malayalam (members of the Dravidian language family). Both tasks are multimodal, consisting of videos paired with files containing just their audio tracks as well as (sometimes partial) text transcripts. For the purposes of this class, we will be working on the second subtask of sentiment analysis in Tamil and Malayalam.

2.1 Primary Task: Sentiment Analysis of Text Data

Our task is to categorize text written in either Tamil or Malayalam into 5 different categories: Highly Negative, Negative, Neutral, Positive, or Highly Positive. The data is ordinal, that is, the categories are discrete and there exists a total order over them, but the distances between the categories are taken to be unknown or not well-defined. In our initial attempts, we tried formulating the task both in terms of ordinal regression and in terms of multiclass classification.

Previous work on sentiment analysis on Dravidian languages has been done using code-switched data (English and either Tamil or Malayalam) (Chakravarthi et al., 2021). Our data is not code-switched, and only contains one language. Like many tasks in Natural Language Processing, different Neural Network architectures have been used to perform sentiment analysis (Habimana et al., 2019).

2.2 Adaptation Task: Sentiment Analysis of Multimodal Data: Text, Audio, and Video

For our adaptation task, we will continue to work on sentiment analysis; however, we will be includ-

[1] The official citation has yet to be released as of May 9, 2023.

ing audio and video data in addition to the text data from the primary task. This task will also be multilingual, using data from both Tamil and Malayalam. The same five categories will be used in the adaptation task as in the primary task.

Work on analyzing and indexing videos has been summarized in Snoek and Worring (2005). Additionally, Dimitri (2022) contains an extensive overview of multimodal integration methods and their applications, including for the purpose of sentiment analysis.

2.3 Data set

2.3.1 Description

The shared task organizers have provided, in total, 52 Tamil samples and 70 Malayalam samples. Each sample consists of a video containing speech in the given language, the corresponding audio file, and a (sometimes partial) transcript of the speech in the audio. The samples are all movie reviews collected from YouTube and labeled by human annotators. The shared task organizers intend to also drop an additional training dataset later on in the timeline of the shared task.

The shared task organizers permit the use of other data sources and pretrained models as long as they are named and cited properly. We are still exploring the possibility of using other data; at the present time, it is unlikely we will do so. We are, however, likely to use pretrained models in some way.

2.3.2 Split

The task organizers have provided a train and dev split over the data set, which is further subdivided by language (Tamil and Malayalam). The 52 Tamil samples are split 42/10 train/dev while the Malayalam samples are split 60/10.

However, instead of using the official split, we combined the train and dev data into a single train set, and then used k-fold validation as we explain later.

For evaluation, an official evaluation dataset will be provided in the coming weeks by the organizers.

The official data can be found in Google Drive folder linked below:

- Tamil train data
- Tamil dev data
- Malayalam train data
- Malayalam dev data

2.4 Evaluation

The organizers of the competition have stated that they will use an F1 metric for evaluation purposes. They have not release evaluation tools or scripts at this time, but suggest teams use Sklearn's classification report function (which lists precision, recall, F1, and a confusion matrix) in evaluating performance, suggesting that they may use the same function in evaluation tools (whether eventually released or not). We intend to follow their advice in this matter.

We note that F1 is a problematic metric for ordinal data, as it is ambivalent to the ordering of the class labels; for example, given a review whose true label is "highly positive," an F1 score would equally penalize a model classifying it as "positive" as "highly negative." However, since this is the metric of choice for the task organizers, we will use it as our baseline evaluation metric (over the official "dev" data, which is our test data). Nevertheless, we are currently exploring other metrics we can use, both for evaluation of our end-to-end system, and for use as a loss metric when training our model(s), which would more accurately capture the inter-class ordering of our labels.

3 System Overview

As you see in Figure 1, there are three main stages to our system pipeline, namely preprocessing/tokenization, vectorization, and K-fold Cross Validation.

First, we preprocess and tokenize the data. To preprocess the data we use two different tokenization methods the data. For the baseline models, we use whitespace tokenization after removing stopwords, punctuation, and numbers. For the more sophisticated models, we use the XLM-RoBERTa (Conneau et al., 2019) tokenizer, which was trained using the SentencePiece algorithm (Kudo and Richardson, 2018). Text longer than the model maximum input length (512, or 510 after accounting for special tokens) is truncated, while text shorter than it is padded.

Second, we then vectorize each input sample to produce document feature vectors. The baseline models use TF-IDF vectorization. The more sophisticated models use last four hidden layers of the XLM-RoBERTa base model. After tokenizing the text, we pass it through the model and get the last four hidden layers' states. Then, we concatenate each of those hidden layer's CLS token represen-

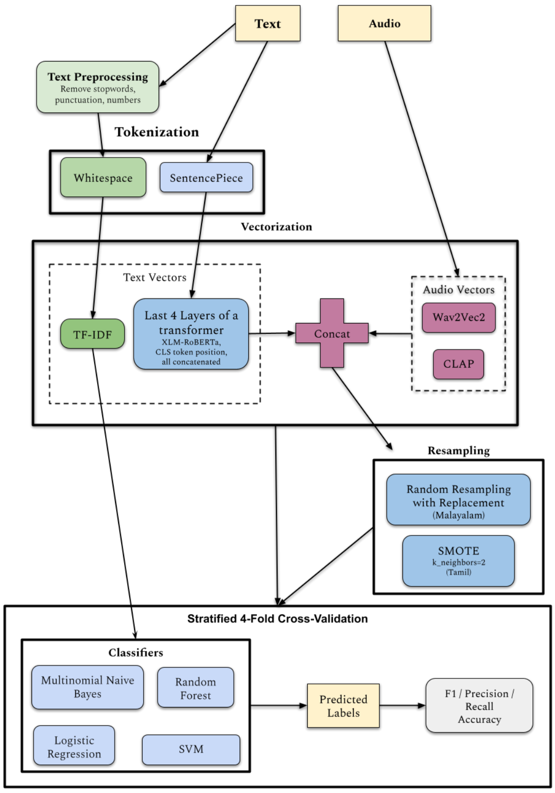

Figure 1: End-to-End System Architecture

tations (the first row in each layer) together. Since XLM-RoBERTa has a hidden feature size of 768, this gives us a 3072-feature document embedding. When analyzing audio data, we concatenate the audio vectors to the text vectors.[2]

Finally, we run stratified k-fold cross validation with $k = 4$. K-fold cross validation consists of two main components: classification and evaluation. For classification, we classify each document into the 5 different sentiment categories. To classify, we first train various models using the training data, and then use the various trained models to predict the sentiment of the development documents specified by the current fold. For evaluation, we evaluate the performance of each model using F_1 score.

4 Approach

4.1 Baseline

This section details the different baseline models that we implement. All baseline models used no resampling, and all the models performed identically to a Majority Class classifier, i.e., they classified all instances as the majority class, POSITIVE. All baseline classifiers except the finetuned transformer LM used the sklearn library (Pedregosa et al., 2011), while the finetuned transformers used HuggingFace (Wolf et al., 2020).

4.1.1 Preprocessing

During preprocessing for the baseline, we first tokenized the data by whitespace. We then removed any tokens containing punctuation or numbers, as well as stop words. We used the list of stop words provided by `spaCy`[3] for both Tamil and Malayalam.

4.1.2 TF-IDF Vectors

We then create one TF-IDF vector per document in the data set. To create the TF-IDF vectors, we run each document through the TF-IDF vectorizer provided by `sklearn`.[4]

To calculate TF-IDF, we used the unsmoothed TF-IDF provided by `sklearn`. Given a document set D with n documents, a document $d \in D$, and a term t with document frequency $df(t)$, we calculate TF-IDF for term t as follows:

$$\text{TFIDF}(t, d, D) = tf(t, d) \cdot \text{idf}(t) \tag{1}$$

$$tf(t, d) = count(t) \in d \tag{2}$$

$$\text{idf}(t, D) = \log\left(\frac{n}{df(t)}\right) + 1 \tag{3}$$

4.1.3 Multinomial Naive Bayes Classifier

We trained a multinomial Naive Bayes classifier on the TF-IDF feature vectors from section 4.1.2. To run Naive Bayes, we used `sklearn`.

4.1.4 Logistic Regression Classifier

We trained a linear Logistic Regression classifier as an additional baseline, also using `sklearn`. We tested different combinations of hyperparemeters including the solver, penalty, regularization strength, and the maximum number of iterations. Although this classifier, under certain (relatively rare) conditions, could replicate the results of the Naive Bayes classifier on default settings, it never outperformed it.

4.1.5 Stochastic Gradient Descent

We trained a Stochastic Gradient Descent classifier as an additional baseline, again using

[2] As of D3, this is a work in progress.

[3] We used spaCy v3.*. The Tamil and Malayalam language models can be found here.

[4] We used sklearn v1.*. `Sklearn`'s TF-IDF vectorizer documentation can be found here

sklearn. We tested different combinations of hyperparameters including alpha, learning rate, and the maximum number of iterations, but it always underperformed the Naive Bayes classifier on default settings, so we do not present the results here.

4.2 Finetuned RoBERTa and Indic Bert Models

We also finetuned the XLM RoBERTa model (available from HuggingFace) on our task. We tried training across a variety of hyperparameter settings: we varied the number of epochs from 5 to as many as 1000, we tried freezing or unfreezing various layers at various points during training, and we tried training with both a multiclass classifier head and a regression head. However, in terms of actual accuracy over the k-fold validation sets, results did not vary.

4.3 XLM-RoBERTa Last Four Layers

Our more sophisticated models vectorized the text. For Tamil, we also resampled the data using SMOTE. We could not use SMOTE for Malayalam due to the fact that there was only a single HIGHLY NEGATIVE sample. These models showed some improvements in the macro F1 score when used with SMOTE but this usually (but not always) came at the cost of accuracy.

5 Results

5.1 Evaluation

To evaluate our classifiers, we pooled the train and dev into one large training set, and then ran k-fold Cross-Validation. This allowed us to train our models on more training data, as there was not a lot of training data provided. For k-fold Cross-Validation, we used 4 folds. To be able to compare the different models that we created, we used a deterministic algorithm to shuffle the data. This ensures that the same four train and dev splits were used for all the models, making the performance of our different models comparable.

For k-fold Cross-Validation, we report the per-fold weighted f_1 score, the per-fold weighted accuracy, and the per-fold weighted precision [5]. We also report the pooled weighted f_1 score, pooled weighted accuracy, and the pooled weighted precision, by taking the averages of the respective per-fold scores over all four folds.

5.2 Baseline

The multinomial Naive Bayes classifier assigns each document the POSITIVE label.

Table 1: Naive Bayes Baseline

Tamil

	Precis.	Accuracy	F1
Fold 0	0.41	0.64	0.50
Fold 1	0.33	0.57	0.42
Fold 2	0.38	0.62	0.47
Fold 3	0.38	0.62	0.47
Average	0.37	0.61	0.46

Malayalam

	Precis.	Accuracy	F1
Fold 0	0.36	0.60	0.45
Fold 1	0.36	0.60	0.45
Fold 2	0.36	0.60	0.45
Fold 3	0.36	0.60	0.45
Pooled	0.36	0.60	0.45

5.3 XLM RoBERTa Model

All permutations of our finetuned XLM RoBERTa model performed the same as the NB classifier, assigning each document the POSITIVE label.

Table 2: Pooled scores for Tamil

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	4
Negative	0.00	0.00	0.00	4
Neutral	0.00	0.00	0.00	6
Positive	0.61	1.00	0.76	33
Highly pos.	0.00	0.00	0.00	7
Accuracy			0.61	54
Macro avg.	0.12	0.20	0.15	54
Weighted avg.	0.37	0.61	0.46	54

Table 3: Pooled scores for Malayalam

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	0
Negative	0.00	0.00	0.00	10
Neutral	0.00	0.00	0.00	6
Positive	0.60	1.00	0.75	36
Highly pos.	0.00	0.00	0.00	7
Accuracy			0.60	60
Macro avg.	0.12	0.20	0.15	60
Weighted avg.	0.36	0.60	0.40	60

[5] For per-fold scores, see Appendix A.1

5.4 Logistic Regression Classifier

Table 4: Logistic Regression (XLM-RoBERTA Last Four Layers)

Tamil (SMOTE)

	Precis.	Recall	F1	Supp.
Highly neg.	0.18	0.50	0.27	4
Negative	0.09	0.25	0.13	4
Neutral	0.33	0.33	0.33	6
Positive	0.64	0.21	0.32	33
Highly pos.	0.20	0.43	0.27	7
Accuracy			0.28	54
Macro avg.	0.29	0.34	0.26	54
Weighted avg.	0.47	0.28	0.30	54

Malayalam

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.00	0.00	0.00	10
Neutral	0.00	0.00	0.00	6
Positive	0.60	1.00	0.75	36
Highly pos.	0.00	0.00	0.00	7
Accuracy			0.60	60
Macro avg.	0.12	0.20	0.15	60
Weighted avg.	0.36	0.60	0.45	60

5.5 Random Forest Classifier

Table 5: Random Forest

Tamil (SMOTE)

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	4
Negative	0.00	0.00	0.00	4
Neutral	0.00	0.00	0.00	6
Positive	0.60	0.91	0.72	33
Highly pos.	0.50	0.14	0.22	7
Accuracy			0.57	54
Macro avg.	0.22	0.21	0.19	54
Weighted avg.	0.43	0.57	0.47	54

Malayalam

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.50	0.10	0.17	10
Neutral	0.00	0.00	0.00	6
Positive	0.60	0.97	0.74	36
Highly pos.	0.00	0.00	0.00	7
Accuracy			0.60	60
Macro avg.	0.22	0.21	0.18	60
Weighted avg.	0.45	0.60	0.47	60

6 Discussion & Error Analysis

In our D2 baselines, without resampling, our classifiers did not perform better than a majority class classifier due to the fact that the amount of data available to train on is fairly small and most of the true labels are POSITIVE. For D3, we found using SMOTE for Tamil increased macro F1. In the future we will look at ways to mitigate this problem.

7 Ethical Considerations

7.1 Data Collection Methods

One must consider the way in which multimodal data is collected. Because multimodal data collection for sentiment analysis involves human participants, one must consider both the privacy and the informed consent of the human participants who engage in the data collection process.

7.1.1 Privacy

Unlike text data, where one can scrape the internet and grab data from an anonymous source, multimodal data uses both audio and video visuals, both which are means of identifying an individual. In the sentiment analysis task, using methods such as disguising a participants voice or hiding their face is undesirable, as both the voice as well as the facial expressions of the individual are informative and can help the model make decisions concerting the participant's sentiment. Therefore, even without associating a certain sample with a certain individual, the individual might still be identified by a family member, friend, or acquaintance.

For video in particular, researchers must also ensure that unwanted background is not visible, such as other people, commodities in the person's home, or items which might help identify the location of the participant. One method might be to blur the video background, but this can negatively influence the generalizability of machine learning models (Schuller et al., 2016).

7.1.2 Informed Consent

Because this task uses modes of data from which a participant can be identified from and anonymity of the participants is nearly impossible to ensure, researches must get informed consent from all participants.

As people act differently when they have knowledge that they are being recorded, it is ideal that participants don't know that they are participating in the data collection process during sample collection time, because then the most organic reactions would be obtained. This has the implication of recording someone first, and then obtaining informed consent later. This method of collection has multitudes of ethical and legal concerns (Schuller et al., 2016).

Therefore, samples must be taken after informed consent takes place, and participants therefore might not have the most genuine and spontaneous responses. This has the implication that the data gathered is not the most accurate portrayal of a participant's sentiment.

7.2 Sentiment Analysis Technology Applications

One concern held by Schuller et al. (2016) is that one would feel that the models used for sentiment analysis as objective, while a model is only as good as the data it was trained on. This has the implication that one must carefully consider the application of the sentiment analysis technology, and the effect that different tasks have when incorrect decisions are inevitably made by the model. For instance, an incorrect sentiment decision made by the personal recommendation system on Netflix would not have a very large implication, while an incorrect sentiment decision made during the President's State of the Union Address might have massive implications.

Another concern held by Schuller et al. (2016), is that computing models which perform sentiment analysis often make simplifying assumptions, thereby reducing the intricacies of human sentiment to a level which is not representative of true human sentiment.

8 Conclusion

References

Premjith B., Sowmya V., Joyithish Lal. G., Bharathi Raja Chakravarthi, K. Nandhini, Rajeswari Natarajan, Abirami Murugappan, and Bharathi B. 2023. Multimodal Abusive Language Detection and Sentiment Analysis:DravidianLangTech@RANLP 2023.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Shardul Suryawanshi, Navya Jose, Elizabeth Sherly, and John P. McCrae. 2021. Overview of the track on sentiment analysis for dravidian languages in code-mixed text. In *Proceedings of the 12th Annual Meeting of the Forum for Information Retrieval Evaluation*, FIRE '20, page 21–24, New York, NY, USA. Association for Computing Machinery.

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

Jingfeng Cui, Zhaoxia Wang, Seng-Beng Ho, and Erik Cambria. 2023. Survey on sentiment analysis: evolution of research methods and topics. *Artificial Intelligence Review*.

Giovanna Maria Dimitri. 2022. A short survey on deep learning for multimodal integration: Applications, future perspectives and challenges. *Computers*, 11(11).

Olivier Habimana, Yuhua Li, Ruixuan Li, Xiwu Gu, and Ge Yu. 2019. Sentiment analysis using deep learning approaches: an overview. *Science China Information Sciences*, 63(1):111102.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Björn Schuller, Jean-Gabriel Ganascia, and Laurence Devillers. 2016. Multimodal sentiment analysis in the wild: Ethical considerations on data collection, annotation, and exploitation. In *Proceedings of the 1st International Workshop on ETHics In Corpus Collection, Annotation and Application (ETHI-CA $ ^2 $2016), satellite of the 10th Language Resources and Evaluation Conference (LREC 2016)(2016)*, pages 29–34.

Cees GM Snoek and Marcel Worring. 2005. Multimodal video indexing: A review of the state-of-the-art. *Multimedia tools and applications*, 25:5–35.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on*

Empirical Methods in Natural Language Processing: System Demonstrations, pages 38–45, Online. Association for Computational Linguistics.

A Appendices

A.1 D2 Results

Table 6: Naive Bayes Baseline

Tamil

	Precis.	Accuracy	F1
Fold 0	0.41	0.64	0.50
Fold 1	0.33	0.57	0.42
Fold 2	0.38	0.62	0.47
Fold 3	0.38	0.62	0.47
Average	0.37	0.61	0.46

Malayalam

	Precis.	Accuracy	F1
Fold 0	0.36	0.60	0.45
Fold 1	0.36	0.60	0.45
Fold 2	0.36	0.60	0.45
Fold 3	0.36	0.60	0.45
Pooled	0.36	0.60	0.45

Table 7: Tamil - XLM RoBERTa Model

Fold 0

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.00	0.00	0.00	1
Neutral	0.00	0.00	0.00	2
Positive	0.64	1.00	0.78	9
Highly pos.	0.00	0.00	0.00	1
Accuracy			0.64	14
Macro avg.	0.13	0.20	0.16	14
Weighted avg.	0.41	0.64	0.50	14

Fold 1

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.00	0.00	0.00	1
Neutral	0.00	0.00	0.00	2
Positive	0.62	1.00	0.76	8
Highly pos.	0.00	0.00	0.00	2
Accuracy			0.57	14
Macro avg.	0.11	0.20	0.15	14
Weighted avg.	0.33	0.57	0.42	14

Fold 2

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.00	0.00	0.00	1
Neutral	0.00	0.00	0.00	1
Positive	0.62	1.00	0.76	8
Highly pos.	0.00	0.00	0.00	2
Accuracy			0.62	13
Macro avg.	0.12	0.20	0.15	13
Weighted avg.	0.38	0.62	0.47	13

Fold 3

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.00	0.00	0.00	1
Neutral	0.00	0.00	0.00	1
Positive	0.62	1.00	0.76	8
Highly pos.	0.00	0.00	0.00	2
Accuracy			0.62	13
Macro avg.	0.12	0.20	0.15	13
Weighted avg.	0.38	0.62	0.47	13

Pooled Scores

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	4
Negative	0.00	0.00	0.00	4
Neutral	0.00	0.00	0.00	6
Positive	0.61	1.00	0.76	33
Highly pos.	0.00	0.00	0.00	7
Accuracy			0.61	54
Macro avg.	0.12	0.20	0.15	54
Weighted avg.	0.37	0.61	0.46	54

Table 8: Malayalam - XLM RoBERTa Model

Fold 0

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	0
Negative	0.00	0.00	0.00	2
Neutral	0.00	0.00	0.00	2
Positive	0.60	1.00	0.75	9
Highly pos.	0.00	0.00	0.00	2
Accuracy			0.60	15
Macro avg.	0.12	0.20	0.15	15
Weighted avg.	0.36	0.60	0.45	15

Fold 1

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.00	0.00	0.00	2
Neutral	0.00	0.00	0.00	1
Positive	0.60	1.00	0.75	9

	Precis.	Recall	F1	Supp.
Highly pos.	0.00	0.00	0.00	2
Accuracy			0.60	15
Macro avg.	0.12	0.20	0.15	15
Weighted avg.	0.36	0.60	0.45	15

Fold 2

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	0
Negative	0.00	0.00	0.00	3
Neutral	0.00	0.00	0.00	1
Positive	0.60	1.00	0.75	9
Highly pos.	0.00	0.00	0.00	2
Accuracy			0.60	15
Macro avg.	0.12	0.20	0.15	15
Weighted avg.	0.36	0.60	0.45	15

Fold 3

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	0
Negative	0.00	0.00	0.00	3
Neutral	0.00	0.00	0.00	2
Positive	0.60	1.00	0.75	9
Highly pos.	0.00	0.00	0.00	1
Accuracy			0.60	15
Macro avg.	0.12	0.20	0.15	15
Weighted avg.	0.36	0.60	0.45	15

Pooled Scores

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.00	0.00	0.00	10
Neutral	0.00	0.00	0.00	6
Positive	0.60	1.00	0.75	36
Highly pos.	0.00	0.00	0.00	7
Accuracy			0.60	60
Macro avg.	0.12	0.20	0.15	60
Weighted avg.	0.36	0.60	0.45	60

A.2 D3 Results

Table 9: Logistic Regression (XLM-RoBERTA Last Four Layers)

Tamil (SMOTE)

	Precis.	Recall	F1	Supp.
Highly neg.	0.18	0.50	0.27	4
Negative	0.09	0.25	0.13	4
Neutral	0.33	0.33	0.33	6
Positive	0.64	0.21	0.32	33
Highly pos.	0.20	0.43	0.27	7
Accuracy			0.28	54
Macro avg.	0.29	0.34	0.26	54
Weighted avg.	0.47	0.28	0.30	54

Malayalam

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.00	0.00	0.00	10
Neutral	0.00	0.00	0.00	6
Positive	0.60	1.00	0.75	36
Highly pos.	0.00	0.00	0.00	7
Accuracy			0.60	60
Macro avg.	0.12	0.20	0.15	60
Weighted avg.	0.36	0.60	0.45	60

Table 10: Random Forest

Tamil (SMOTE)

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	4
Negative	0.00	0.00	0.00	4
Neutral	0.00	0.00	0.00	6
Positive	0.60	0.91	0.72	33
Highly pos.	0.50	0.14	0.22	7
Accuracy			0.57	54
Macro avg.	0.22	0.21	0.19	54
Weighted avg.	0.43	0.57	0.47	54

Malayalam

	Precis.	Recall	F1	Supp.
Highly neg.	0.00	0.00	0.00	1
Negative	0.50	0.10	0.17	10
Neutral	0.00	0.00	0.00	6
Positive	0.60	0.97	0.74	36
Highly pos.	0.00	0.00	0.00	7
Accuracy			0.60	60
Macro avg.	0.22	0.21	0.18	60
Weighted avg.	0.45	0.60	0.47	60