CUET_NetworkSociety@DravidianLangTech 2025: A Transformer-Driven Approach to Political Sentiment Analysis in Tamil X (Twitter) Comments

Tofayel Ahmmed Babu[*], **Sabik Aftahee**[*], **MD Musa Kalimullah Ratul**[*]
Jawad Hossain and Mohammed Moshiul Hoque
Department of Computer Science and Engineering
Chittagong University of Engineering and Technology
{u1904005, u1904024, u1904071, u1704039}@student.cuet.ac.bd
moshiul_240@cuet.ac.bd

Abstract

Social media has become an established medium of public communication and opinions on every aspect of life, but especially politics. This has resulted in a growing need for tools that can process the large amount of unstructured data that is produced on these platforms providing actionable insights in domains such as social trends and political opinion. Low-resource languages like Tamil present challenges due to limited tools and annotated data, highlighting the need for NLP focus on understudied languages. To address this, a shared task has been organized by DravidianLangTech@NAACL 2025 for political sentiment analysis for low-resource languages, with a specific focus on Tamil. In this task, we have explored several machine learning methods such as SVM, AdaBoost, GB, deep learning methods including CNN, LSTM, GRU BiLSTM, and the ensemble of different deep learning models, and transformer-based methods including mBERT, T5, XLM-R. The mBERT model performed best by achieving a macro F1 score of 0.2178 and placing our team 22$^{\text{nd}}$ in the rank list.

1 Introduction

Understanding and interpreting human emotions and opinions expressed in text has become a vital aspect of natural language processing (NLP), particularly in the context of social media and public discourse. With the arrival of social media networks such as X (Twitter), the need for advanced sentiment analysis tools, especially politically, is once more highlighted. Political sentiment analysis can provide key insights into the opinion of the population, party identification, and social concerns which are of great interest to policymakers, analysts, and political operators. However, it is difficult to find an advanced sentiment analysis tool for any low-resource languages such as Tamil (Chen et al., 2015). Tamil is a speech of more than 80 million people worldwide (Jain et al., 2020), and therefore a source of a huge corpus of internet dialogue, especially social media. However, Tamil is poorly represented in NLP literature due to the morphological complexity and lack of annotated datasets. Efforts to create annotated datasets for Tamil sentiment analysis (Chakravarthi, 2020), along with tools for morphological analysis (Sarveswaran et al., 2021), have provided a solid foundation for further advancements in the field. Additionally, the development of hybrid architectures that combine deep learning techniques (Ramesh Babu, 2022) with multilingual transformer models (Roy and Kumar, 2021) has shown considerable promise in addressing the challenges posed by the complex linguistic nature of Tamil. However, despite these strides, the lack of substantial recent advancements in comprehensive toolkits for Tamil NLP continues to hinder progress, especially in specialized tasks such as political sentiment analysis. This paper aims to build on these foundational efforts by proposing an improved system for political sentiment analysis in Tamil tweets. Our contributions include:

- Developed a transformer-based system for political multiclass sentiment analysis of Tamil X (Twitter) comments.
- Investigated various machine learning, deep learning, and transformer-based models for Tamil political sentiment analysis and conducted an in-depth error analysis to evaluate the performance of these models.

2 Related Work

Recent advancements in sentiment analysis for low-resource languages have increasingly relied on multilingual transformers such as XLM-R (Conneau, 2019) and IndicBERT (Kannan et al., 2021), which have demonstrated effectiveness in Tamil

[*]Authors contributed equally to this work.

Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages, pages 614–620
May 3, 2025

sentiment classification. Integrating contextualized embeddings (Kenton and Toutanova, 2019; Liu, 2019) and transfer learning techniques (Ruder et al., 2019) has further addressed the challenges of data scarcity. Additionally, CNN-based architectures (Kim, 2014) continue to contribute to feature extraction, reinforcing the role of deep learning in sentiment analysis for low-resource languages. Nazir et al. (2025) proposed a transformer-based approach using CMSA-mBERT for multiclass sentiment analysis (Positive, Negative, and Neutral) on the CMDSA-24 dataset, achieving F1 score of 79.87% and the result shows huge improvements over traditional methods. Khan et al. (2025) proposed an attention-based, stacked CNN-BiLSTM model for Urdu sentiment analysis, improving feature extraction and sequential pattern recognition. Evaluated on UCSA-21 and UCSA datasets, it achieved an accuracy of 83.12% and 78.91%, respectively.

For Tamil-specific NLP tasks, Sarveswaran et al. (2021) created tools specifically for Tamil morphology, providing a baseline for further research. Jain et al. (2020) developed a high-quality Tamil-to-English translation system that outperforms Google Translator (which might indirectly be useful for sentiment analysis problems) and its effects on text representation. Attai et al. (2024) provided a useful insight into political discourse and analysis. They used machine learning techniques (SVM, RF, XGBoost) to analyze public sentiment in the 2023 Nigerian General Elections and it was established that 43% of the tweets were Neutral, 33% Positive, and 24% Negative, with XGBoost achieving the highest accuracy of 93%. Sampath and Supriya (2024) explored sentiment analysis on code-mixed data using translation-based preprocessing and transformer models, achieving 94% accuracy with DistilBERT for Tamil-English and 92% for Hindi-English. Results highlight the effectiveness of specialized NLP models over traditional translation tools. Moreover, a study by K et al. (2023) on textual sentiment analysis in Tamil and Tulu code-mixed texts employed SVM and ensemble models with fastText and TF-IDF, achieving F1-scores of 0.14 (Tamil) and 0.20 (Tulu).

3 Task and Dataset Description

In the shared task, the provided dataset (Chakravarthi et al., 2025) comprises three CSV files for training, validation, and testing, containing 4352, 544, and 544 data points, respectively. Notably, the dataset exhibits class imbalance across its sentiment categories. The dataset includes seven sentiment classes: *Opinionated, Sarcastic, Neutral, Positive, Substantiated, Negative,* and *None of the Above*. Detailed class-wise statistics are presented in Table 1.

Class	**Train**	**Val**	**Test**	$\mathbf{W}_T$	$\mathbf{UW}_T$
Opinionated	1361	153	171	31748	13540
Sarcastic	790	115	106	17231	8717
Neutral	687	84	70	13975	7075
Positive	575	69	75	13251	6459
Substantiated	412	52	51	10310	5679
Negative	406	51	47	9079	4997
None of the Above	171	20	25	1619	1193
Total	**4352**	**544**	**544**	**97213**	**47660**

Table 1: Class-wise distribution of train, validation, and test set for political multiclass sentiment analysis of Tamil X (Twitter) comments, where val, W_T, and UW_T denote validation, total words in each class, and total unique words in each class, respectively

For enhanced visualization, bar charts have been included in the Appedix B. The implementation details of the tasks will be found in the GitHub repository[1].

4 Methodology

Various machine learning (ML), deep learning (DL), and transformer-based models were utilized to create a strong baseline, as shown in Figure 1.

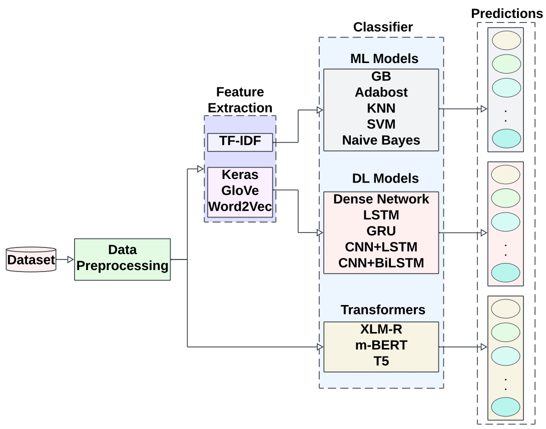

Figure 1: Schematic process of political multiclass sentiment analysis of Tamil X (Twitter) comments

4.1 Data Preprocessing

The dataset contained 4352, 544, and 544 samples for the training, validation, and testing sets, re-

[1] https://github.com/5pace4/NAACL-2025

spectively. To ensure data integrity, instances with missing values in the content or labels columns were removed. Preprocessing was primarily aimed at standardization with the use of the unidecode library that aimed to remove the accents, lower the text to set it to default, and clean up the non-alphanumerical and digital forms with regular expression, removing non-alphabet texts and numbers and conversion to lowercase. Other preprocessing steps included clearing double or more spaces into one and paragraph formatting. Then sentiment labels are converted to numerical values using *LabelEncoder* so that the machine learning model can handle data efficiently.

4.2 Feature Extraction

Feature extraction in NLP transforms raw text into machine-readable numerical representation through various techniques. It varied for machine learning, deep learning, and transformer-based models in this paper. Machine learning models (SVM, GB, AdaBoost, KNN, Naive Bayes) used TF-IDF vectorization with unigram and bigram features, mapping 5,440 samples (4,352 training, 544 validation, 544 test) into 5,000-dimensional matrices, yielding shapes of (4,352 × 5,000) for training and (544 × 5,000) for validation and test, respectively. Deep learning models utilized TensorFlow Keras Tokenizer for tokenization and pre-trained embeddings like word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) (glove.6B.300d.txt), forming a (5,001 × 300) embedding matrix, including an OOV token. Transformer models (mBERT, XLM-R, T5) employed contextualized word embeddings with sub-word tokenization, dynamically adjusting representations for sentiment classification. This multilayered feature extraction enhances model efficiency and accuracy.

4.3 Machine Learning Models

Various machine learning algorithms were used to classify sentiment in Tamil such as SVM, GB, AdaBoost, and KNN. SVM finds a separate hyperplane, while GB reduces bias variation, and AdaBoost improves performance by iterating on weak learners. KNN performs the classification of samples based on their proximity in feature space. All models were trained using features extracted with TF-IDF, providing a strong baseline for the classification of Tamil sentiment.

4.4 Deep Learning Models

The proposed approach applies deep learning methods to model both local and global textual features for the sentiment classification of Tamil X (Twitter) comments. This study tries many variants of architecture such as Dense Networks, LSTM, GRU, CNN+LSTM, and CNN+BiLSTM. The Dense Network with Batch Normalization served as the baseline model, integrating fully connected layers with dropout and batch normalization to mitigate overfitting. LSTM and BiLSTM captured long-term dependencies in text sequences, while CNN extracted local patterns using convolutional filters. Additionally, pre-trained embeddings (GloVe and Word2Vec) were explored, serving as input layers for LSTM, GRU, and CNN architectures, which were fine-tuned during training. All models were trained on tokenized and padded text sequences, ensuring uniform input dimensions across different architectures. Table 2 provides the hyperparameters used for deep learning models in sentiment classification, including LSTM, GRU, CNN+LSTM, and CNN+BiLSTM.

Model	Embedding	Layers/Units	Epochs	Batch	Opt.
Dense Net	None	Dense: 256, 128	15	64	Adam
LSTM	GloVe	LSTM: 128, 64	20	32	Adam
GRU	Word2Vec	GRU: 128, 64	15	32	Adam
CNN+LSTM	GloVe	Conv1D: 128, LSTM: 64	15	32	Adam
CNN+LSTM	Word2Vec	Conv1D: 128, LSTM: 64	15	32	Adam
CNN+BiLSTM	Word2Vec	Conv1D: 128, BiLSTM: 64	15	32	Adam
CNN+BiLSTM	GloVe	CNN: 256, BiLSTM: 128, 64	30	32	Adam
GRU	GloVe	GRU: 128, 64	15	32	Adam

Table 2: Hyperparameters of deep learning for sentiment classification

The models use either GloVe or Word2Vec embeddings and are trained with the Adam optimizer. Batch sizes range from 32 to 64, while epochs vary between 15 and 30, ensuring a balance between computational efficiency and model performance.

4.5 Transformer-Based Models

Transformer-based models including mBERT (Devlin, 2018), XLM-R (Conneau, 2019), and T5 (Ni et al., 2021) were fine-tuned for sentiment classification in Tamil text. All these models are based on pre-trained contextual embeddings that capture subtle semantic nuances. Of these, the best macro F1 score is obtained by mBERT, benefiting from its robust multilingual training. The cross-lingual task-optimized XLM-R gave competitive results, while T5 effectively handled structured outputs using a sequence-to-sequence approach. Fine-tuning involved task-specific adaptations and optimizing cross-entropy loss. Each model has been tuned based on a set of several hyperparameters summarised in Table A.1 Appendix A.

5 Result Analysis

This paper introduced the political sentiment classification of Tamil text. The performance of sentiment classification models was evaluated using precision, recall, and macro F1 score across ML, DL, and transformer-based approaches. A summarizing table of various models performance is shown in Table 3. Among the machine learning models, SVM achieved the best F1 score, which is 0.2167. The strength of the SVM lies in its ability to find a separating hyperplane between sentiment classes in a high-dimensional space. AdaBoost got a slightly lower F1 score of 0.2088, while Gradient Boosting (GB) achieved F1 score of 0.2028. However, KNN and NB demonstrated limited performance, achieving F1 scores of 0.1430 and 0.1045, respectively, due to their simplicity and inability to model complex patterns in the dataset effectively.

Classifiers	Precision	Recall	F1 Score
SVM	0.3345	0.2547	0.2167
GB	0.3476	0.2620	0.2028
AdaBoost	0.2288	0.2581	0.2088
KNN	0.2339	0.2203	0.1430
Naive Bayes	0.4360	0.1592	0.1045
Deep Learning Models			
Dense Network	0.1968	0.1637	0.1270
LSTM (G)	0.2636	0.1946	0.1917
CNN+LSTM (G)	0.2743	0.1863	0.1840
CNN+LSTM (W)	0.3032	0.1971	0.1887
CNN+BiLSTM (G)	0.3151	0.2160	0.2166
CNN+BiLSTM (W)	0.3339	0.1991	0.1891
GRU (G)	0.2903	0.1876	0.1868
GRU (W)	0.2671	0.1822	0.1600
Transformer-Based Models			
XLM-R	0.1704	0.1752	0.1317
T5	0.3071	0.2073	0.1937
mBERT	**0.2557**	**0.3190**	**0.2178**

Table 3: Performance comparison of classifiers across ML, DL, and transformer models where G and W denote GloVe and Word2Vec embedding, respectively

In the DL domain, CNN+BiLSTM with GloVe embeddings yielded the highest F1 score of 0.2166, while CNN+BiLSTM with Word2Vec achieved 0.1891. A simpler Dense network had the lowest F1 score of 0.1270, showcasing the limitations of shallow architectures for this task. Other architectures, such as CNN+LSTM with Glove, CNN+LSTM with Word2Vec, GRU with GloVe, GRU with Word2Vec, and LSTM with GloVe achieved F1 scores of 0.1840, 0.1887, 0.1868, 0.1600, and 0.1917, respectively.

The best performance of the transformer-based models was that of mBERT, pre-trained on multilingual datasets with dynamic contextual embeddings, which really worked for the Tamil text, with F1 score of 0.2178 and placed the team 22nd in the final rank list. T5 follows next because of its sequence-to-sequence learning approach, with F1 score of 0.1937. XLM-R obtained the worst F1 score, reaching just 0.1317 probably for being more cross-lingual transfer-focused than fine-tuned for sentiment classification in some languages. Overall, the results show that hybrid deep learning models and transformer-based approaches are immensely better in capturing the rich semantics of Tamil sentiment compared to traditional machine learning methods. However, the modest macro F1 scores across all methods highlight the challenges posed by low-resource languages like Tamil, such as data scarcity and morphological complexity.

6 Error Analysis

Both quantitative and qualitative error analyses were conducted to gain deeper insights into the performance of the proposed model.

6.1 Quantitative Analysis:

The best-performing models were used to conduct a quantitative error analysis, utilizing confusion matrices shown in Figure 2.

The confusion matrix describes performance and challenges concerning seven classes of sentiments in Tamil i.e. *Negative, Neutral, None of the Above, Opinionated, Positive, Sarcastic,* and *Substantiated.* It is noticed that the model has performed well in identifying the *Opinionated* class, followed by *None of the Above* and *Sarcastic*, with the highest number of correct predictions 129 instances, 21, and 19, respectively. However, there were significant misclassification patterns, especially for the *Negative* class, which was almost entirely misclassified, mostly as *Opinionated* with 36 instances. The *Neutral* class was quite confused with *Opinionated,* with 52 instances, showing that there is some difficulty in distinguishing these classes. Though the *Positive* class had 9 correct predictions, a large number of instances were misclassified as *Opinionated,* showing that there is some overlap in semantic features. Moreover, the *Substantiated* class also suffered, with zero correct predictions and heavy misclassifications into

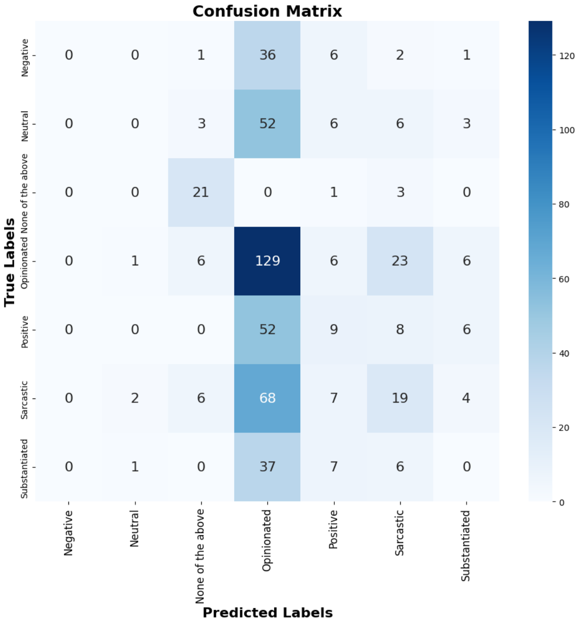

Figure 2: Confusion matrix of the proposed model (fine-tuned mBERT) for political multiclass sentiment analysis

Opinionated and *Positive,* reflecting ambiguities in contextual cues. A major source of these errors seems to be the class imbalance in the dataset. *Opinionated* class dominating others and such imbalance prohibits the model from learning fine-grained features of minority classes, leading to over-reliance on more frequent categories.

Sample Text	Actual Label	Predicted Label
Sample 1: இஸ்லாமிய சகோதரர்களுடன் ரமலான் கொண்டாடிய அதிமுக வேட்பாளர் ராயபுரம் மனோ #royapurammano #admk #chennai #electioncampaign	Neutral	Opinionated
Sample 2: ஓபிஎஸ் - எடப்பாடி போட்டா போட்டி! திடீரென பணிகளை முடுக்கியுள்ள எடப்பாடி! #AIADMK #OPS #EPS #Annamalai #Edappadi #OPanneerselvam #OPSvsEPS #DMK #BJP #Seeman #NaamTamilar #MKStalin #IPL #IPL2023 #CSK #ChennaiSuperKings #RCBVSLSG #ViratvsGambhir #ViratKohli	Negative	Opinionated
Sample 3: mony kathir அரியாத ஜனங்கள்	None of The Above	None of The Above
Sample 4: நன்றி அண்ணா. #மக்களின்_சின்னம்_மைக்	Opinionated	Opinionated

Figure 3: Few examples of predicted outputs by the proposed method (mBERT) for political multiclass sentiment analysis

6.2 Qualitative Analysis:

Figure 3 illustrates the predicted outputs of the proposed model for Tamil political multiclass sentiment analysis based on sample inputs. The model correctly classified Samples 3 and 4 but misclassified *Neutral* (Sample 1) and *Negative* (Sample 2) as *Opinionated,* likely due to contextual bias from hashtags and data imbalance.

7 Conclusion

This study evaluated multiple approaches for classifying political multiclass sentiment in X (Twitter) comments, including ML, DL, and transformer-based models. Among them, mBERT achieved the best macro F1 score (0.2178), benefiting from multilingual pretraining and dynamic contextual embeddings. Hybrid deep learning models, such as CNN+BiLSTM with GloVe, also performed competitively, effectively capturing both local and sequential features. In contrast, traditional ML models struggled with the task's complexity. The overall low F1 scores highlight the challenges of sentiment analysis in a morphologically rich, low-resource language like Tamil. In future work, we aim to mitigate class imbalance using resampling techniques and weighted loss functions, conduct ablation studies to analyze mBERT's performance and explore improvements through data augmentation, fine-tuning, and Tamil-specific preprocessing strategies.

Limitations

Despite the contributions of the current work on political multiclass sentiment analysis of Tamil X (Twitter) comments has several drawbacks. i) As the proposed approach relies on pre-trained transformer-based model, its performance may degrade in scenarios where the context significantly deviates from the data on which the model was originally trained. ii) The focus on Tamil-specific sentiment analysis limits the applicability of the models to other low-resource languages without significant adaptation. iii) The dataset used is imbalanced, which may have impacted the model's ability to generalize across all sentiment categories.

Acknowledgments

We thank the DravidianLangTech 2025 shared task organizers for running this task. This work was supported by the Directorate of Research & Extension (DRE), Chittagong University of Engineering & Technology (CUET).

References

K Attai, D Asuquo, KE Okonny, AB Johnson, A John, I Bardi, C Iroanwusi, and O Michael. 2024. Sentiment analysis of twitter discourse on the 2023 nigerian general elections. *European Journal of Computer Science and Information Technology*, 12(4):18–35.

Bharathi Raja Chakravarthi. 2020. Hopeedi: A multilingual hope speech detection dataset for equality, diversity, and inclusion. In *Proceedings of the Third Workshop on Computational Modeling of People's Opinions, Personality, and Emotion's in Social Media*, pages 41–53.

Bharathi Raja Chakravarthi, Saranya Rajiakodi, Elizabeth Sherly, Thenmozhi Durairaj, Sathiyaraj Thangasamy, Ratnasingam Sakuntharaj, Prasanna Kumar Kumaresan, Kishore Kumar Ponnusamy, Arunaggiri Pandian Karunanidhi, and Rohan R. 2025. Overview of the Shared Task on Political Multiclass Sentiment Analysis of Tamil X(Twitter) Comments: DravidianLangTech@NAACL 2025. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*. Association for Computational Linguistics.

Nancy Chen, Chongjia Ni, I-Fan Chen, Sunil Sivadas, Tung Pham, Haihua Xu, Xiong Xiao, Tze Lau, Su-Jun Leow, Boon Pang Lim, Cheung-Chi Leung, Chin-Hui Lee, Alvina Goh, Eng Chng, Bin Ma, and Haizhou Li. 2015. Low-resource keyword search strategies for tamil. volume 2015.

A Conneau. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Minni Jain, Ravneet Punia, and Ishika Hooda. 2020. Neural machine translation for tamil to english. *Journal of Statistics and Management Systems*, 23(7):1251–1264.

Rachana K, Prajnashree M, Asha Hegde, and H. L Shashirekha. 2023. MUCS@DravidianLangTech2023: Sentiment analysis in code-mixed Tamil and Tulu texts using fastText. In *Proceedings of the Third Workshop on Speech and Language Technologies for Dravidian Languages*, pages 258–265, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

R Ramesh Kannan, Ratnavel Rajalakshmi, and Lokesh Kumar. 2021. Indicbert based approach for sentiment analysis on code-mixed tamil tweets. In *FIRE (Working Notes)*, pages 729–736.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1. Minneapolis, Minnesota.

Lal Khan, Atika Qazi, Hsien-Tsung Chang, Mousa Alhajlah, and Awais Mahmood. 2025. Empowering urdu sentiment analysis: an attention-based stacked cnn-bi-lstm dnn with multilingual bert. *Complex & Intelligent Systems*, 11(1):10.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Preprint*, arXiv:1408.5882.

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Muhammad Kashif Nazir, CM Nadeem Faisal, Muhammad Asif Habib, and Haseeb Ahmad. 2025. Leveraging multilingual transformer for multiclass sentiment analysis in code-mixed data of low-resource languages. *IEEE Access*.

Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith B Hall, Daniel Cer, and Yinfei Yang. 2021. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. *arXiv preprint arXiv:2108.08877*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Suba Sri Ramesh Babu. 2022. *Sentiment Analysis In Tamil Language Using Hybrid Deep Learning Approach*. Ph.D. thesis, Dublin, National College of Ireland.

Pradeep Kumar Roy and Abhinav Kumar. 2021. Sentiment analysis on tamil code-mixed text using bi-lstm. In *FIRE (Working Notes)*, pages 1044–1050.

Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Tutorials*, pages 15–18.

Koyyalagunta Krishna Sampath and M Supriya. 2024. Transformer based sentiment analysis on code mixed data. *Procedia Computer Science*, 233:682–691.

Kengatharaiyer Sarveswaran, Gihan Dias, and Miriam Butt. 2021. Thamizhimorph: A morphological parser for the tamil language. *Machine Translation*, 35:1–34.

Model	Tokenizer	Learning Rate	Epochs	Batch Size	Max Length	Warmup Steps
mBERT	WordPiece	5e-6	17	8	256	500
XLM-R	Byte Pair Encoding (BPE)	1e-6	20	8	256	500
T5	SentencePiece	2e-4	12	8	256	500

Table A.1: Hyperparameters of transformer-based models for sentiment classification

A Tuned Hyperparameters

Table A.1 lists the hyperparameters used in transformer-based models, such as mBERT, XLM-R, and T5. Their differences include tokenization methods, learning rates, and training configurations. The batch size, maximum sequence length, and warm-up steps of all transformer models are 8, 256, and 500, respectively, to maintain stable learning. These tables together present a comparative view of the experimental setup that enables understanding of how different deep learning and transformer architectures were fine-tuned for Tamil sentiment classification.

B Class Distribution

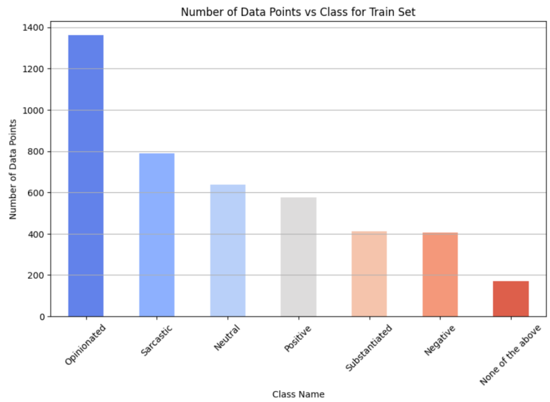

Figure B.1: Number of datapoints of each class in train dataset

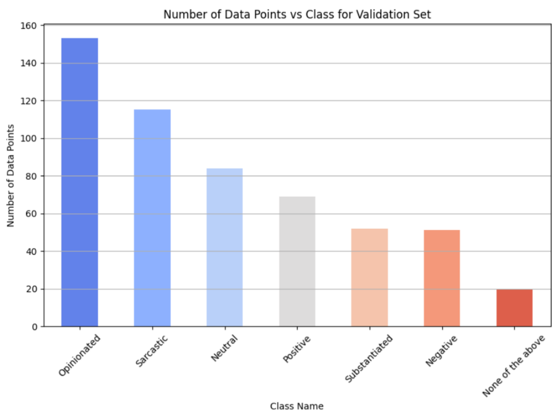

Figure B.2: Number of datapoints of each class in validation dataset

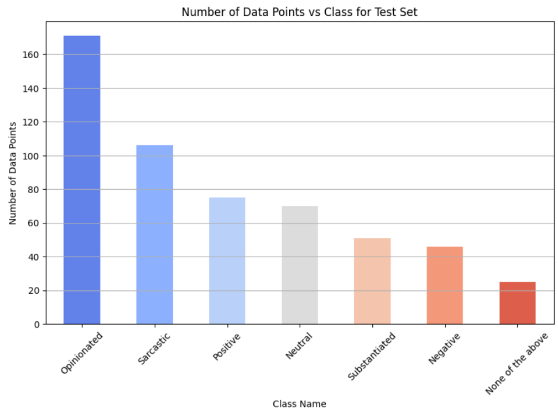

Figure B.3: Number of datapoints of each class in test dataset

The Figures B.1, B.2, and B.3 demonstrate the number of data points for each class in the training, validation, and test set, respectively.