Wise@LT-EDI-2025: Combining Classical and Neural Representations with Multi-scale Ensemble Learning for Code-mixed Hate Speech Detection

Ganesh Sundhar S[1], Durai Singh K[1], Gnanasabesan G[1], Hari Krishnan N[1], Dhanush MC[1]
[1]Amrita School of Artificial Intelligence, Coimbatore
Amrita Vishwa Vidyapeetham, India
{cb.en.u4aie22017, cb.en.u4aie22167, cb.en.u4aie22018, cb.en.u4aie22020, cb.en.u4aie22130}
@cb.students.amrita.edu

Abstract

Detecting hate speech targeting caste and migration communities in code-mixed Tamil-English social media content is challenging due to limited resources and socio-cultural complexities. This paper proposes a multi-scale hybrid architecture combining classical and neural representations with hierarchical ensemble learning. We employ advanced preprocessing including transliteration and character repetition removal, then extract features using classical TF-IDF vectors at multiple scales (512, 1024, 2048) processed through linear layers, alongside contextual embeddings from five transformer models-Google BERT, XLM-RoBERTa (Base and Large), SeanBenhur BERT, and IndicBERT. These concatenated representations encode both statistical and contextual information, which are input to multiple ML classification heads (Random Forest, SVM, etc). A three-level hierarchical ensemble strategy combines predictions across classifiers, transformer-TF-IDF combinations, and dimensional scales for enhanced robustness. Our method scored an F1-score of 0.818, ranking 3rd in the LT-EDI-2025 shared task, showing the efficacy of blending classical and neural methods with multi-level ensemble learning for hate speech detection in low-resource languages.

Keywords: Caste/Migration-based hate speech detection, Code-mixed text, Transliteration, TF-IDF Features, Transformer embeddings, Hierarchical ensemble learning, Low-resource languages

1 Introduction

Hate speech is any kind of communication that attacks a person or group based on attributes like caste, religion, race, or other identity factors. With the advancement of technology and the advent of social media, individuals can now share their thoughts with anyone in the world. While this increased connectivity has many benefits, the anonymity offered by online platforms has unfortunately facilitated the spread of hateful messages, particularly those targeting vulnerable groups such as migrants and specific caste communities. To make online communities inclusive, it is essential to identify caste and migration-based hate speech.

Despite significant progress in Natural Language Processing(NLP) through transformer architectures (Vaswani et al., 2017) revolutionizing text classification tasks, detecting hate speech in low-resource languages like Tamil poses unique challenges stemming from dialectal diversity and regional variations. This challenge is further intensified when detecting specific forms of hate speech, such as those targeting caste and migration, as these are often embedded in local socio-cultural nuances and context. In social media platforms, the texts are often code-mixed, i.e., English + Tamil, making detection even more challenging.

To address these challenges, our work presents a multi-scale hybrid architecture that combines classical Term Frequency-Inverse Document Frequency(TF-IDF) (Spärck Jones, 1972) features at multiple scales (512, 1024, and 2048) with contextual transformer embeddings. For final classification, a hierarchical ensemble using majority voting across models and feature scales was used to make the model robust and generalizable.

2 Related Work

Early hate speech detection systems relied on rule-based methods and keyword matching (Clarke et al., 2023). These methods are unsuitable for the vast amount of data present today as they lack contextual understanding. Machine learning algorithms like Support Vector Machines (Kp et al., 2009) and Logistic Regression (Hosmer Jr et al., 2013), which used hand-crafted features such as TF-IDF, n-grams, and Parts of Speech (POS) tags, emerged later. Although these models improved

Proceedings of the Fifth Workshop on Language Technology for Equality, Diversity, Inclusion, pages 54–62

performance, they were ineffective for code-mixed or culture-specific hate speech (Davidson et al., 2017).

Deep learning techniques such as RNNs (Elman, 1990) and LSTMs (Hochreiter and Schmidhuber, 1997), when combined with word embeddings like Word2Vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), improved context-aware modeling and reduced the need for manual feature engineering (Pitsilis et al., 2018). Nevertheless, they struggled with long-range dependencies and noisy, code-mixed text. With the emergence of pre-trained language models such as BERT (Devlin et al., 2019a), RoBERTa (Liu et al., 2019), and multilingual variants like XLM-R (Conneau et al., 2020a), the performance improved further. Fine-tuning these pretrained models on hate speech datasets has consistently yielded better results (Albladi et al., 2025).

In their work, (Roy et al., 2022) used an ensemble-based approach to detect hate speech in code-mixed Tamil and Malayalam texts, as the individual models had a high misclassification rate. Two ensemble techniques were used: one based on the average of the outcomes and another using custom weights. Their ensemble model outperformed the previously reported state-of-the-art models, achieving an F1 score of 0.933 on Tamil and 0.802 for the Malayalam dataset.

In their work, (Sreelakshmi et al., 2024) detected Hate Speech and Offensive Language (HOS) in low-resource Dravidian CodeMix languages (Kannada, Malayalam, Tamil). Various multilingual transformer-based embeddings (e.g., MuRIL, BERT, XLM-R) were combined with traditional ML classifiers for HOS detection. To address class imbalance, a cost-sensitive learning approach was used. Experiments on six datasets showed that MuRIL + SVM performed best overall.

3 Task and Dataset Description

The goal of this shared task (Rajiakodi et al., 2025) is to develop a system to detect hate speech targeting caste and migrant communities in code-mixed social media data, focusing on Tamil, a low-resource language. The dataset (Rajiakodi et al., 2024) has three columns: "text", which has the comments from social media platforms; "id", containing the ID of the comments; and "label", which is set to 1 for hate speech and 0 for non-hate speech. The dataset description is provided in Table 1.

Dataset	No. of comments
Train	5512
Dev	787
Test	1576
Total	7875

Table 1: Distribution of comments across training, development, and test sets

4 Methodology

The approach uses a multi-scale hybrid framework to identify hate speech in code-mixed Tamil social media posts. Two preprocessing schemes (with and without transliteration) are used to handle intrinsic noise in the data, such as emojis, URLs, inconsistent spacing, repeated characters in transliterated Tamil words, and code-mixing between Tamil and English languages. Feature extraction unites TF-IDF vectors at three dimensions (512, 1024, 2048) with contextual embeddings of five transformer models. Such features are then concatenated to create integrated feature vectors with both statistical and contextual information. A set of 22 traditional ML classifiers are trained for each feature set, and the top 3 models for each were chosen. A three-level hierarchical ensemble approach employs majority voting across classifiers, feature sets, and dimensions to ensure resilient classification by combining heterogeneous preprocessing schemes, representation types, and model architectures

4.1 Data Preprocessing

Our data pre-processing pipeline addressed the challenges of social media text containing code-mixed Tamil and English content. In the first approach, without transliteration, newlines were replaced with white spaces, emojis were converted into text i.e. demojization (Kim and Wurster, 2014), URLs were removed, multiple whitespaces were replaced with a single space, and then the text was converted into lowercase.

In the second approach, the same steps were repeated, followed by transliteration (Karimi et al., 2011) of Tamil Unicode characters into their English equivalents, and repeated characters in transliterated Tamil words were removed (while preserving standard English words), and then non-ASCII characters were removed to maintain consistency.

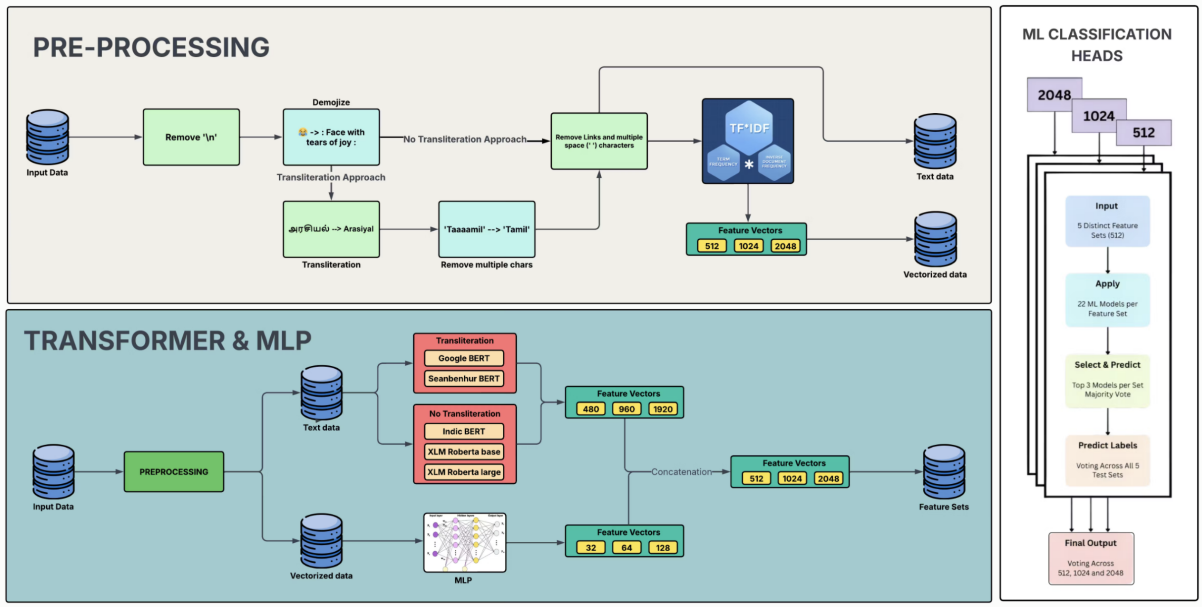

Figure 1: Multi-scale architecture for hate speech detection: Three-stage pipeline showing (a) data preprocessing with dual approaches (with/without transliteration) producing both processed text data and TF-IDF vectors, (b) transformer models generating contextual embeddings that are combined with reduced TF-IDF features to create unified Wise Embeddings (WE), and (c) machine learning classifiers processing WE features followed by three-level hierarchical ensemble voting to produce final hate speech predictions.

4.2 TF-IDF Vectorization

TF-IDF vectorization (S N et al., 2022) was used to extract features from the text, and its hyperparameters were optimized using grid search (Hutter et al., 2019). This resulted in high-dimensional feature vectors consisting of approximately 22,000 features. As these feature vectors are sparse, Truncated Singular Value Decomposition (SVD) was applied:

$$\mathbf{X} \approx \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T \quad (1)$$

This decomposition (Halko et al., 2011) was used to reduce the feature space to three different dimensions: 512, 1024, and 2048. These features were further refined using Feed-Forward Networks (FFNs) (Rumelhart et al., 1986) to produce compact embeddings of 32, 64, and 128 dimensions respectively.

4.3 Transformer Embeddings

Five different transformer models were used to extract contextual embeddings from the preprocessed text. For the models Google BERT (Devlin et al., 2019b) and SeanBenhur BERT (Benhur and Sivanraju, 2021), the input was text preprocessed with transliteration, while for Indic BERT (Kp et al., 2025), XLM RoBERTa base (Conneau et al., 2020b), and XLM RoBERTa large, the input was text preprocessed without transliteration. The contextual representations were extracted from the [CLS] token, which serves as an aggregate representation of the entire input sequence. The transformer models produce embeddings of varying dimensions: IndicBERT, XLM RoBERTa Base, and SeanBenhur BERT generate 768-dimensional embeddings, while Google BERT Large and XLM RoBERTa Large produce 1024-dimensional embeddings. These obtained embeddings undergo linear transformation to achieve target dimensions of 480, 960, and 1920 for the three scales respectively. Each transformed embedding is then concatenated with its corresponding FFN-reduced feature vectors (of dimensions 32, 64, and 128), resulting in unified representations of 512, 1024, and 2048 dimensions that capture both statistical and contextual information. The unified representations obtained are hereafter referred to as Wise Embeddings (WE).

4.4 Machine Learning Models

For each of the five feature sets at every dimension in WE, 22 traditional machine learning classifiers including Logistic Regression, SVM, Naive Bayes (McCallum and Nigam, 1998), and Random Forest Classifier (Breiman, 2001) were trained. From these, the top three classifiers were selected based on their validation performance.

For each feature set and scale, predictions from the top three classifiers were aggregated using majority voting to give a single predicted label. Then, the five resulting predictions for each dimension (5 BERT models) was again combined using majority voting to produce a final label per dimension. Then, a cross-dimensional ensemble was performed, where the three labels i.e. one from each WE dimension underwent another round of majority voting to determine the overall predicted label.

During the initial transformer training phase, the Wise Embeddings (WE) are processed through a final linear layer that maps the 512, 1024, or 2048-dimensional representations to a single output value, optimized using Binary Cross-Entropy (BCE) (Goodfellow et al., 2016) loss. This linear layer effectively functions as a linear classifier, constraining the learned WE representations to be linearly separable in the feature space. However, the complex nature of code-mixed hate speech detection often exhibits non-linear patterns that cannot be adequately captured by linear decision boundaries alone.

By applying traditional ML classifiers with inherently non-linear decision boundaries (such as Random Forest and SVM with non-linear kernels) to these linearly-optimized WE features, we introduce additional modeling capacity to capture complex patterns in the data. This approach leverages the pre-trained linear separability while allowing non-linear classifiers to model intricate relationships that the original linear layer could not capture. The ensemble voting across multiple classifiers further enhances robustness and generalization, particularly beneficial for handling the noisy and heterogeneous nature of code-mixed social media text.

5 Result and Analysis

The model's performance was assessed using the F1-score (Powers, 2011), which is defined in Equation 2.

$$F1_{macro} = \frac{1}{C}\sum_{i=1}^{C}\frac{2 \times P_i \times R_i}{P_i + R_i} \quad (2)$$

The results of the best-performing models for different dimensions (512, 1024, and 2048) are summarized in Table 2 and the results of different ensembles are given in Table 3. The ensemble of the three different dimensions achieved the highest F1-score of 0.85 in the dev set.

Dim	Transformer	Best Model	F1
512	Google BERT	RF	0.80
	IndicBERT	RF	0.81
	SeanBenhur BERT	XGBoost	0.81
	XLM-R Base	RF	0.81
	XLM-R Large	Extra Trees	0.81
1024	Google BERT	SVM	0.81
	IndicBERT	RF	0.83
	SeanBenhur BERT	RF	0.82
	XLM-R Base	Nu-SVM	0.83
	XLM-R Large	RF	0.83
2048	Google BERT	Gradient Boosting	0.84
	IndicBERT	Ridge Regression	0.82
	SeanBenhur BERT	Nu-SVM	0.83
	XLM-R Base	RF	0.84
	XLM-R Large	RF	0.83

Table 2: PerformancemMetrics of the combinations

Ensemble Type	F1-score
512 dim models	0.81
1024 dim models	0.82
2048 dim models	0.84
Cross-dimensional ensemble	0.85

Table 3: Performance metrics of multi-scale ensembles

5.1 Comparison

Our proposed methodology using Wise Embeddings (WE), achieved an F1-score of 0.81827 on the test set, securing 3rd rank in the shared task. The F1 scores of the top five performing teams in the shared task are summarized in Table 4.

Rank	Team Name	F1-score
1	CUET_N317	0.88105
2	CUET's_white_walkers	0.86289
3	Wise	0.81827
4	CUET_blitz_aces	0.81682
5	hinterwelt	0.80916

Table 4: Top 5 Teams ranked based on F1-score

6 Conclusion

The present work demonstrates the strength of combining neural and conventional representations through a multi-level ensemble approach for caste and migration-based hate speech detection in code-mixed Tamil text. The approach highlights the importance of employing diverse feature representations in addressing challenging NLP problems in low-resource languages.

Github Source code: https://github.com/Ganesh2609/CasteMigrationHateSpeech

7 Limitations

Even though the proposed pipeline performed well, there are a few limitations, which are as follows:

1. The relatively small dataset (7,875 comments in total) may limit the model's ability to generalize across the full spectrum of hate speech variations in Tamil social media content.
2. Data inconsistency exists where identical comments appear in both training and development sets with conflicting labels, potentially compromising the model's learning process and evaluation reliability.
3. Label quality issues are present in the dataset, where some clearly hateful content is labeled as non-hate speech, while certain benign comments are marked as hate speech. This annotation ambiguity, which is challenging even for human annotators, introduces noise that may affect model performance.

References

Aish Albladi, Minarul Islam, Amit Das, Maryam Bigonah, Zheng Zhang, Fatemeh Jamshidi, Mostafa Rahgouy, Nilanjana Raychawdhary, Daniela Marghitu, and Cheryl Seals. 2025. Hate speech detection using large language models: A comprehensive review. *IEEE Access*, 13:20871–20892.

Sean Benhur and Kanchana Sivanraju. 2021. Pretrained transformers for offensive language identification in tanglish. *arXiv preprint arXiv:2110.02852*.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Christopher Clarke, Matthew Hall, Gaurav Mittal, Ye Yu, Sandra Sajeev, Jason Mars, and Mei Chen. 2023. Rule by example: Harnessing logical rules for explainable hate speech detection. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 364–376, Toronto, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *Preprint, arXiv:1703.04009*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. 2013. *Applied logistic regression*. John Wiley & Sons.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. *Automated machine learning: methods, systems, challenges*. Springer Nature.

Sarvnaz Karimi, Falk Scholer, and Andrew Turpin. 2011. Machine transliteration survey. *ACM Computing Surveys (CSUR)*, 43(3):1–46.

Taehoon Kim and Kevin Wurster. 2014. emoji: Emoji for python. https://github.com/carpedm20/emoji.

Soman Kp, Rajendran Loganathan, and Ajay Vadakkepatt. 2009. *Machine learning with SVM and other kernel methods*.

Suriya Kp, Durai Singh K, Vishal A S, Kishor S, and Sachin Kumar S. 2025. Synapse@DravidianLangTech 2025: Multi-class political sentiment analysis in Tamil X (Twitter) comments: Leveraging feature fusion of IndicBERTv2 and lexical representations. In *Proceedings of the Fifth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 716–720, Acoma, The Albuquerque Convention Center, Albuquerque, New Mexico. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint, arXiv:1907.11692*.

Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48. AAAI Press.

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

Jeffrey Pennington, Richard Socher, and Christopher 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, Doha, Qatar. Association for Computational Linguistics.

Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48(12):4730–4742.

David Martin Powers. 2011. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1):37–63.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Sathiyaraj Thangasamy, Bhuvaneswari Sivagnanam, and Charmathi Rajkumar. 2024. Overview of Shared Task on Caste and Migration Hate Speech Detection. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity and Inclusion*, Malta. European Chapter of the Association for Computational Linguistics.

Saranya Rajiakodi, Bharathi Raja Chakravarthi, Rahul Ponnusamy, Shunmuga Priya MC, Prasanna Kumar Kumaresan, Sathiyaraj Thangasamy, Bhuvaneswari Sivagnanam, Balasubramanian Palani, Kogilavani Shanmugavadivel, Abirami Murugappan, and Charmathi Rajkumar. 2025. Findings of the shared task on caste and migration hate speech detection. In *Proceedings of the Fifth Workshop on Language Technology for Equality, Diversity and Inclusion*, Italy. Fifth Conference on Language, Data and Knowledge (LDK2025).

Pradeep Kumar Roy, Snehaan Bhawal, and Chinnaudayar Navaneethakrishnan Subalalitha. 2022. Hate speech and offensive language detection in dravidian languages using deep ensemble framework. *Computer Speech Language*, 75:101386.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Prasanth S N, R Aswin Raj, Adhithan P, Premjith B, and Soman Kp. 2022. CEN-Tamil@DravidianLangTech-ACL2022: Abusive comment detection in Tamil using TF-IDF and random kitchen sink algorithm. In *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, pages 70–74, Dublin, Ireland. Association for Computational Linguistics.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21. Reprinted in Journal of Documentation, Vol. 60, No. 5, pp. 493-502, 2004.

K. Sreelakshmi, B. Premjith, Bharathi Raja Chakravarthi, and K. P. Soman. 2024. Detection of hate speech and offensive language codemix text in dravidian languages using cost-sensitive learning approach. *IEEE Access*, 12:20064–20090.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

A Training Performance Metrics

This appendix presents the training and validation performance metrics for all transformer models across the three dimensional scales (512, 1024, and 2048). Each figure shows the loss, accuracy, and F1-score curves during the training process.

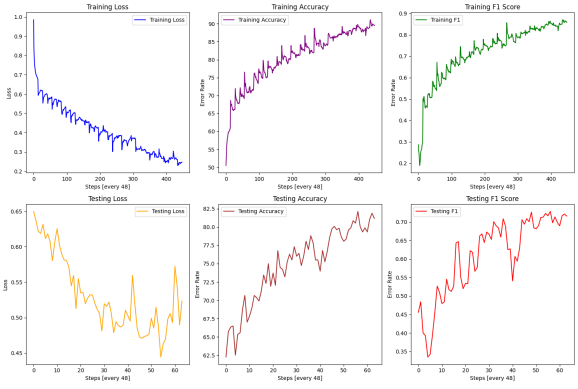

Figure 2: Training and validation metrics for Google BERT with 512-dimensional embeddings.

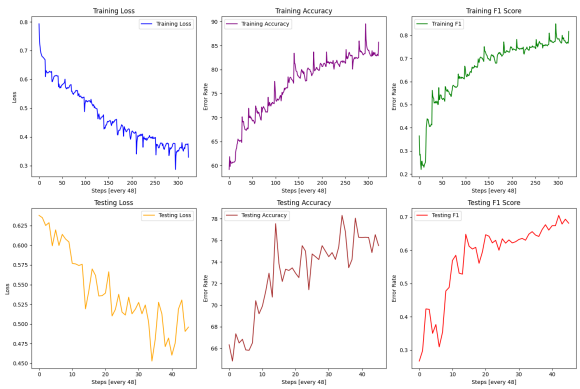

Figure 3: Training and validation metrics for IndicBERT with 512-dimensional embeddings.

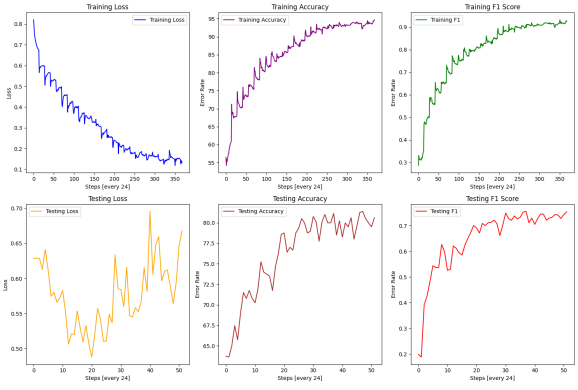

Figure 4: Training and validation metrics for SeanBenhur BERT with 512-dimensional embeddings.

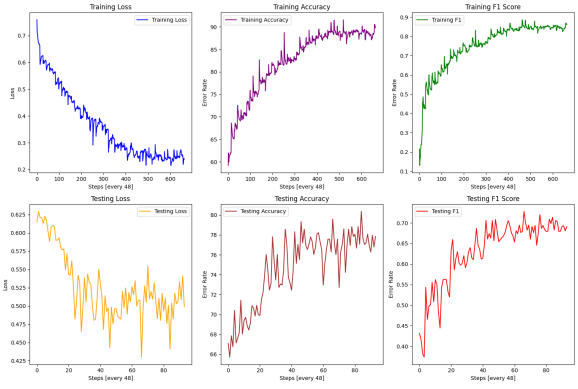

Figure 5: Training and validation metrics for XLM-RoBERTa Base with 512-dimensional embeddings.

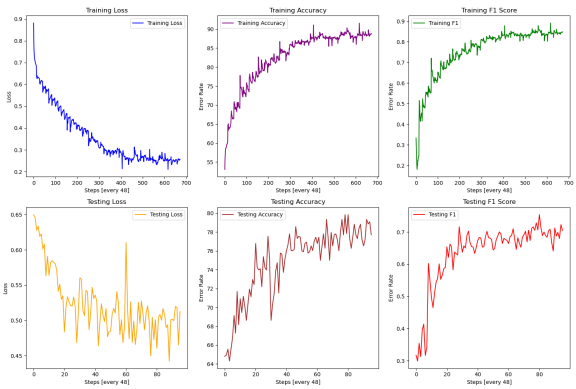

Figure 6: Training and validation metrics for XLM-RoBERTa Large with 512-dimensional embeddings.

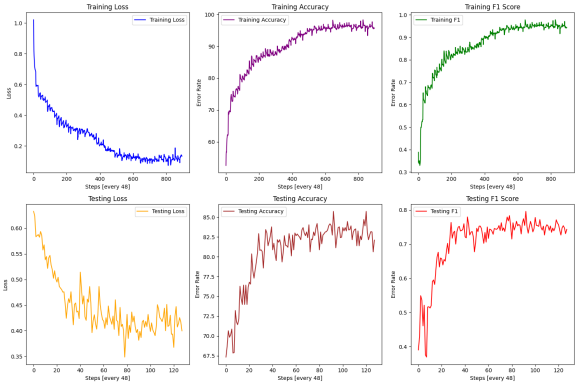

Figure 7: Training and validation metrics for Google BERT with 1024-dimensional embeddings.

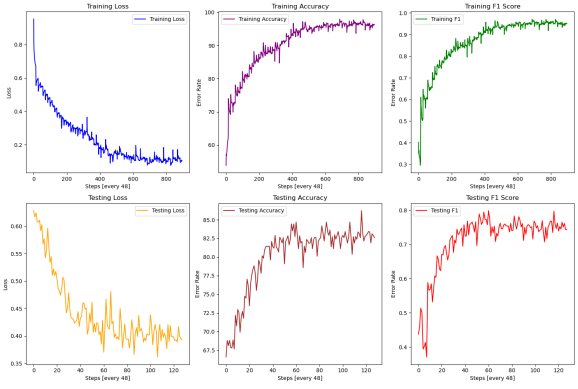

Figure 8: Training and validation metrics for IndicBERT with 1024-dimensional embeddings.

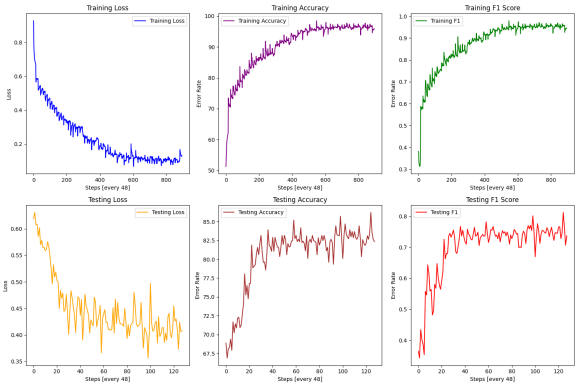

Figure 9: Training and validation metrics for SeanBenhur BERT with 1024-dimensional embeddings.

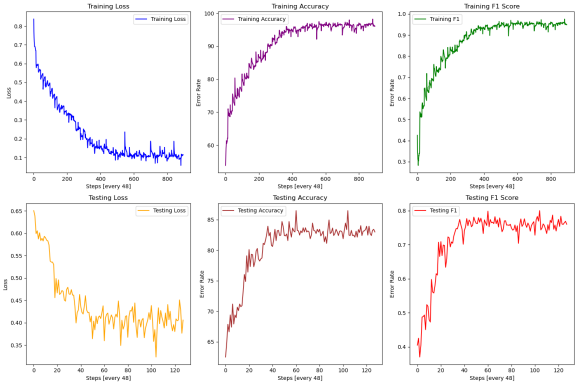

Figure 10: Training and validation metrics for XLM-RoBERTa Base with 1024-dimensional embeddings.

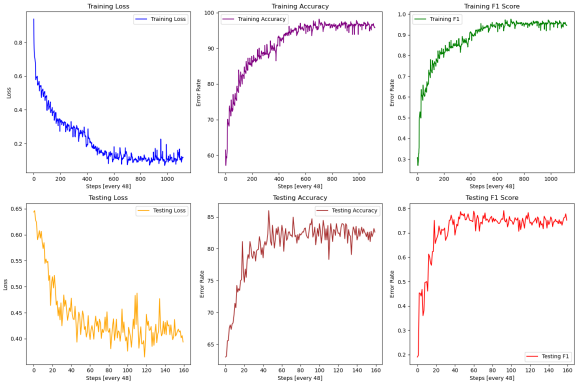

Figure 11: Training and validation metrics for XLM-RoBERTa Large with 1024-dimensional embeddings.

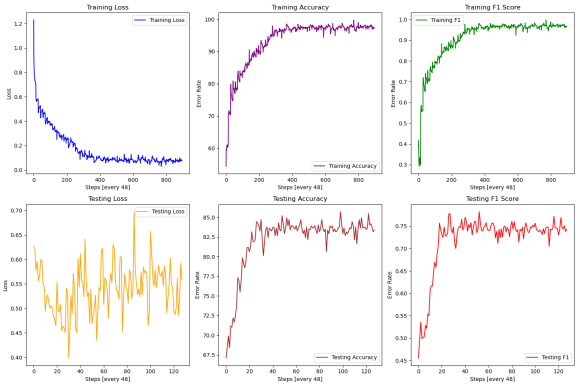

Figure 12: Training and validation metrics for Google BERT with 2048-dimensional embeddings.

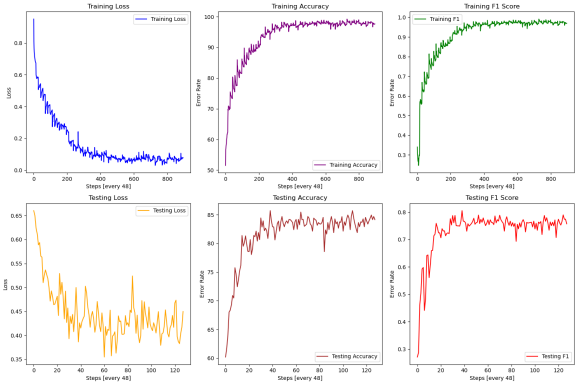

Figure 13: Training and validation metrics for IndicBERT with 2048-dimensional embeddings.

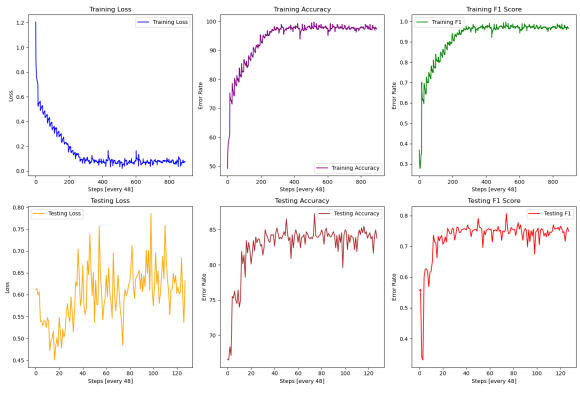

Figure 14: Training and validation metrics for SeanBenhur BERT with 2048-dimensional embeddings.

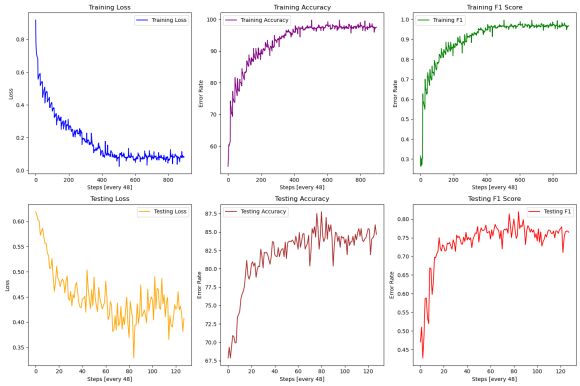

Figure 15: Training and validation metrics for XLM-RoBERTa Base with 2048-dimensional embeddings.

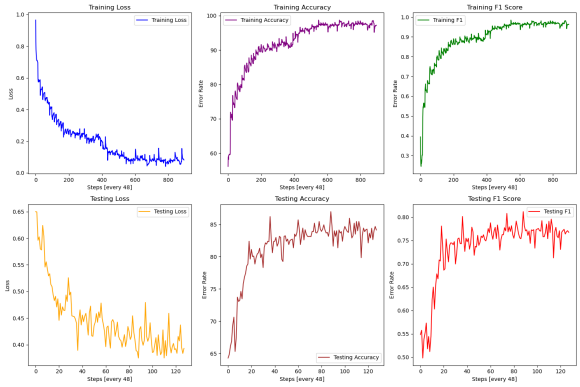

Figure 16: Training and validation metrics for XLM-RoBERTa Large with 2048-dimensional embeddings.